DESIGNING DIGITAL CONSTRUCTIVE VISUALIZATION TOOLS

Gonzalo Gabriel Méndez

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews

2018

Designing Digital Constructive Visualization Tools

Gonzalo Gabriel Méndez

University of
St Andrews

This thesis is submitted in partial fulfillment for the degree of

Doctor of Philosophy

at the University of St Andrews

April 2018

Abstract

The emergence of tools that support fast and easy creation of visualizations has made the benefits of Information Visualization (InfoVis) more accessible. The predominant design for visualization authoring tools often includes features such as automated mappings and visualization templates, which make tools effective and easy-to-use. These features, however, still impose barriers to non-experts (i.e., people with no formal training on visualization concepts). The paradigm of Constructive Visualization (ConstructiveVis) has shown potential to overcome some of these barriers, but it has only been investigated through the use of physical tokens that people manipulate to create representations of data.

This dissertation investigates how the principles of ConstructiveVis can be applied in the design and implementation of digital constructive visualization tools. This thesis presents the results of several observational studies that uncover how tools that promote a constructive approach to visualization compare to more conventional ones. It also sheds light on what kind of benefits and limitations digital ConstructiveVis brings into non-experts' visualization design process.

The investigations here presented lay the foundations for the design of better visualization tools that not only allow people to create effective visualizations but also promote critical reflection on design principles.

Acknowledgements

During my time in St Andrews, I have had the fortune to meet and work with amazing people from whom I have learned invaluable lessons. Hereby, I would like to thank them all.

To Dr Miguel A. Nacenta, supervisor of my research, for his support and guidance throughout these years. I am grateful to you for helping me develop research but, more importantly, for supporting me to develop myself as a researcher. Thanks for your countless advice on how to conduct rigorous academic research and for your constant willingness to share bits of wisdom that always made things better.

Thanks to the awesome crew of the SACHI lab. In particular to Dr Uta Hinrichs, for being always encouraging and for opening the doors of qualitative research to me. Thanks also to Prof. Aaron Quigley, for always finding room in your busy schedule to share great ideas. And to my good friend Shyam Reyal, for making St Andrews feels like home.

Thanks also to all the people who participated in the studies reported in this dissertation. HCI research would be impossible without people who are willing to dedicate a few minutes (or a few hours) of their time to advance science.

Thanks to the friends who supported me during these years from places all over the world. For being *there*, for the moments you shared with me both in person and over the distance, and for making me smile. Special thanks to Raquel Méndez, Letty Rizzo, María Fernanda Vargas, Diana Jaramillo, Luis Galárraga, Alberto Cremades, Daniele Fonte, and Lea Atzinger.

Thanks to my colleagues from the CTI-ESPOL. To Dr Katherine Chiluiza and Dr Enrique Peláez, for your permanent support during the very initial stages of my research career. And special thanks to Dr Xavier Ochoa, for mentoring me and for teaching me that to do research is *just* "answering questions".

Finally, I would like to thank my family. None of this would have been possible without your love, support, encouragement, and prayers. I am infinitely lucky to have you all.

To my parents
Susana y Domingo

Declaration

Candidate's Declarations

I, Gonzalo Gabriel Méndez, hereby certify that this thesis, which is approximately 40,000 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student and as a candidate for the degree of Doctor of Philosophy in April, 2014; the higher study for which this is a record was carried out in the University of St Andrews between 2014 and 2017.

Date:

Signature of candidate:

Supervisor's Declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of Doctor of Philosophy in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date:

Signature of supervisor:

Permission for Electronic Publication

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the electronic publication of this thesis:

> *Access to printed copy and electronic publication of thesis through the University of St Andrews.*

Date:

Signature of candidate:

Signature of supervisor:

Data Management

Where data has been collected during any evaluation, case study or experiment, the data is being managed according to the permission agreed to by the participants through the consent forms. This generally means that the collected data is only available to the researchers directly involved in the evaluation due to the potentially private and sensitive nature of the collected data (most commonly images of people's likeness). The results of and findings from the evaluations are presented in the relevant parts of the thesis.

In accordance with the participants' consent, all data collected from the participants is stored securely within a locked office and will be deleted or safely deposed of within three (3) years of collection. This includes collected images, questionnaire and answer sheets or any other data provided by the participants.

Publications

Some of the work described in this dissertation has been published previously. The following lists give an overview of these publications.

Articles in Peer-reviewed Conference Proceedings

- **Gonzalo Gabriel Méndez**, Miguel A. Nacenta, and Sebastien Vandenheste. 2016. *iVoLVER: Interactive Visual Language for Visualization Extraction and Reconstruction*. In Proceedings of the 2016 SIGCHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA. DOI=http://dx.doi.org/10.1145/2858036.2858435
- **Gonzalo Gabriel Méndez**, Uta Hinrichs, and Miguel A. Nacenta. 2017. *Bottom-up vs. Top-down: Trade-offs in Efficiency, Understanding, Freedom and Creativity with InfoVis Tools*. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). ACM, New York, NY, USA. DOI=http://dx.doi.org/10.1145/3025453.3025942
- **Gonzalo Gabriel Méndez**, Miguel A. Nacenta and Uta Hinrichs. *Considering Agency and Data Granularity in the Design of Visualization Tools*. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA. DOI=http://dx.doi.org/10.1145/3173574.3174

Extended Abstracts from Demos & Doctoral Consortia

- **Gonzalo Gabriel Méndez** and Miguel A. Nacenta. 2016. *Constructing Interactive Visualizations with iVoLVER*. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16). ACM, New York, NY, USA, 3727-3730. DOI=http://dx.doi.org/10.1145/2851581.2890243
- **Gonzalo Gabriel Méndez**. 2016. *Tools for Opportunistic Information Visualization: Visual Analysis with Non-traditional Data Sources*. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Cambridge, United Kingdom, 2016, pp. 250-251. DOI=http://dx.doi.org/10.1109/VLHCC.2016.7739701
- **Gonzalo Gabriel Méndez** and Miguel A. Nacenta. 2016. *Opportunistic Visualization with iVoLVER*. IEEE Symposium on Visual Languages and

Human-Centric Computing (VL/HCC), Cambridge, United Kingdom, 2016, pp. 234-235. DOI=http://dx.doi.org/10.1109/VLHCC.2016.7739693

- Miguel A. Nacenta and **Gonzalo Gabriel Méndez**. 2017. *iVoLVER: A Visual Language for Constructing Visualizations from In-the-Wild Data*. Interactive Surfaces and Spaces (ISS '17), Brighton, United Kingdom, 2017, pp. 438-441. DOI=https://doi.org/10.1145/3132272.3132299 Best demo jury award.

iVoLVER in Masters Dissertations and Student Projects

- **iVoLVR**[1]: Pre-commercialization, Usability Improvements and Future Productization. Sebastien Vandenheste. 2015. MSc in Computing and Information Technology, University of St Andrews.
- **iVoLVER** and its Potential Use as an Education Tool for Basic Mechanics. Alexandra Johnston. 2017. Senior Honours Project, University of St Andrews.
- Google Cloud Vision API Extension to the **iVoLVER** Visualization Interface Yifeng Jiang. 2017. MSc in Information Technology, University of St Andrews.
- Visualisation of Markov Chain Monte Carlo Statistical Analysis via the **iVoLVER** Library. Alice Lynch. 2017. MSc in Advanced Computer Science, University of St Andrews.

[1]This was the early name of iVoLVER

Online Resources

The production version of iVoLVER and its source code are available at `http://ivolver.cs.st-andrews.ac.uk`. This address also provides access to the experimental material used in the studies reported in this dissertation (e.g., videos, datasets). Each study description points to the specific version of iVoLVER it involved as well as its source code.

Funding

This thesis and the research described herein were mainly supported by the Ecuadorian Government, through a SENESCYT scholarship. The Escuela Superior Politécnica del Litoral and the School of Computer Science of the University of St Andrews also provided financial support for this work. Wacom Co., Ltd. donated the display used in the early stages of the research presented in this dissertation.

Contents

List of Figures

List of Tables

Acronyms

ConstructiveVis Constructive Visualization

CV Computer Vision

HCI Human-Computer Interaction

IDE Integrated Development Environment

InfoVis Information Visualization

iVoLVER **i**nteractive **V**isual **L**anguage for **V**isualization **E**xtraction and **R**econstruction

UI User Interface

VDFP Visual Dataflow Programming

VP Visual Programming

VPL Visual Programming Language

Chapter One

Introduction

What we see changes what we know. What we know changes what we see.

— Jean Piaget (attributed)

In a world of ever-increasing information, existing visualization tools allow people—from different backgrounds and with varied levels of expertise—to explore, analyze, and communicate data. A myriad of these tools[1] are free, easy to use, and even available through the web; all of which has benefited people beyond data experts and visualization designers. Ultimately, this has contributed to democratize visualization technologies [213], a long-standing goal of the Information Visualization (InfoVis) community.

Providing fast and easy ways to build visualizations is a common goal of most existing authoring tools. To this end, tools often implement features that ease and speed up the visualization design process, such as templates, automated design, and recommendation systems (e.g., [25, 69, 156, 176, 221]). These features can be beneficial for several types of designers and scenarios. For example, they often avoid the need of textual programming, making tools more appealing and accessible.

Features like the ones mentioned above have led to a predominant visualization authoring strategy in which users can heavily rely on the tool for many steps of the visualization design process (e.g., those that require some level of expertise, such as deciding visual mappings). Although this type of visualization authoring strategy is probably the most popular and widespread nowadays, previous research has shown that it still imposes some barriers, for example, for novices [71]. It is still unclear whether there exist better ways to design visualization tools that provide more appropriate support for specific types of designers such as casual users or

[1] For the remaining of this dissertation, unless stated otherwise, the term "tool" means "visualization tool".

non-experts (i.e., people with no formal training on visualization concepts). This question is relevant, as democratizing visualization should not be simply about reaching a broader audience, but also about providing specific types of users with the right support.

Along these lines, the field of Constructive Visualization (ConstructiveVis) [93, 95] has explored alternative ways for empowering non-experts in their use of visualization. Rooted in education and learning theories, ConstructiveVis seems promising, for example, to overcome the barriers that non-experts encounter when implementing visualizations in more conventional tools. As a paradigm for visualization authoring, ConstructiveVis has been explored with tangible tokens that people manipulate to produce physical representations of data (e.g., [94, 97]). However, although a visualization process that involves tangible elements has its own appeal, it might not be practical for real-world analytical scenarios that involve digital datasets. **The question about whether Constructive Visualization can be implemented in digital visualization tools remains unanswered.**

1.1 Motivating Scenario

Andrew, who has no formal instruction in visualization design, recently started to track his fitness activity using a smartwatch that synchronizes the collected information with his phone. A mobile application allows Andrew visualize his data and keep a record of his routines and goals. The app also allows downloading the data it collects in the form of comma-separated values files. Using this feature, Andrew decides to further explore his data in a few visualization tools that he has heard of. He initially explores MS Excel and then moves onto Tableau Desktop. Soon, Andrew feels limited in the kind of visualizations he can create with these tools.

In a recent conversation with a friend who happens to be a visualization expert, Andrew learns that representing data with physical tokens (i.e., taking a constructive approach) is generally more flexible and expressive. He decides to give this a try but eventually realizes that given the amount of data he has, a tangible representation of his dataset would not be feasible. After a bit of research, he finds a digital tool that supports a constructive approach to visualization authoring. The tool lets Andrew visualize his data in a less restricted visualization environment and gain interesting insights on himself.

The research questions addressed by this research investigate how people like Andrew (i.e., non-experts in visualization design) can benefit from the use constructive visualization in digital tools.

1.2 Research Questions

This doctoral dissertation redresses the gap described at the beginning of this chapter in the current design space of InfoVis authoring tools. More specifically, it addresses the following questions:

- ***Q1:*** Is it possible to design and implement digital constructive visualization tools? If so,

  Q1.A: What kind of principles could drive the design of such tools?

  Q1.B: What challenges arise in the design and development processes?

- ***Q2:*** How do digital constructive tools compare to others implementing a more conventional visualization authoring strategy?

  Q2.A: What kind of benefits do constructive principles bring into people's visualization design process?

  Q2.B: What limitations do they impose?

  Q2.C: What strategies could be applied to overcome these limitations?

By answering these questions, this research furthers our understanding of how constructive visualization principles can be integrated in digital tools. As a consequence, this dissertation provides tools designers with guidance on how to design better visualization authoring tools by incorporating elements of Constructive Visualization.

1.3 Context

This dissertation is set in the context of three broad areas (Figure 1.1): Interaction Design, InfoVis, and Visual Programming (VP). The latter two use visual representations with two different purposes, respectively: to represent data by encoding it in visual properties, and to support the construction of programs graphically. Within these two areas, most of my research is located at the intersection of

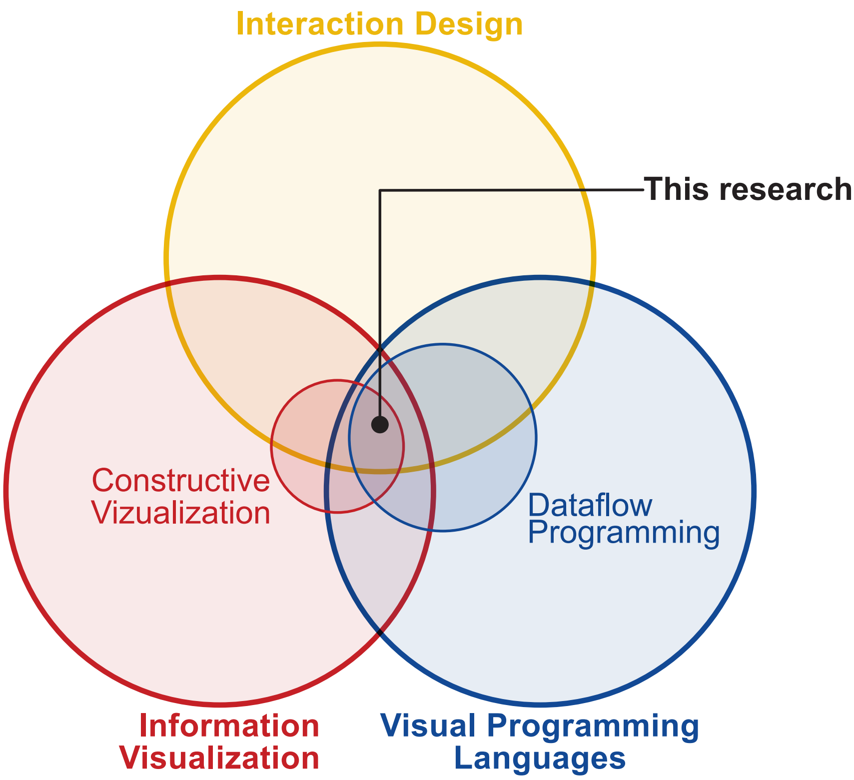


Figure 1.1: Context and scope of the research addressed in this dissertation.

ConstructiveVis and Visual Dataflow Programming (VDFP). ConstructiveVis motivates the problem this thesis addresses: the design, implementation and study of digital constructive visualization tools. On the other hand, the subfield of VDFP supports the implementation of the solution that this thesis contributes and studies: a visual programming environment for visualization that supports a constructive approach.

1.4 Scope

The main focus of this thesis is to investigate how ConstructiveVis principles can be integrated in digital visualization tools and how constructive visualization processes differ from those of more conventional tools. This research focuses on investigating the benefits and limitations of ConstructiveVis regarding the application of visualization design principles and how users approach the visualization design process. However, this dissertation is not concerned, for example, with whether ConstructiveVis tools help people understand the data better or whether they lead to the generation of richer insights.

Additionally, my explorations in the area of digital constructive visualization tools are supported by visual programming (VP). However, I acknowledge that the visual approach I take in this research is arguably just one of many possible ways in which a digital constructive tool can be implemented. That is, there might exist other implementations that do not require the use of visual programming and that are not investigated in this dissertation.

It is also important to highlight that although the initial stages of my research involved the use of pen-and-touch devices, this dissertation does not focus on the area of multi-touch or gesture-based interaction. Also, although I discuss some issues and technical limitations I encountered throughout the execution of this research, I do not intend to provide a comprehensive analysis or description of such issues as they might not generalize to other tool designs beyond the one described in this dissertation.

Finally, given that this is exploratory work, all the experiments reported in this dissertation involved small datasets. Although I do believe the findings from my research are generalizable to interaction with real-world bigger datasets, further research will be needed to provide deeper insights in the area of ConstructiveVis for big data.

1.5 Research Methodology

The work described in this thesis aims at exploring a so far unknown design space of visualization tools. To uncover the elements that compose this space and gain an understanding of their benefits and limitations, this dissertation applies several qualitative research methods.

First, I address the design of a digital constructive visualization tool with a combination of interviews to potential users and an iterative approach [33, 147]. Second, I study this tool from different perspectives through two types of controlled laboratory experiments: a usability study and a two-part observational experiment with non-experts. Finally, inspired in previous work in the area of participatory design [142, 184] and, more specifically, focus groups [111, 112, 114] I present a series of design workshops with people from different backgrounds and expertise levels to dive deeper into the design space of digital constructive visualization tools.

This combination of qualitative data collection methods motivated and evolved ideas throughout the execution of this research. Although other complementary methods could have been used in particular stages of my research, the ones reported above seemed most appropriate given the exploratory nature of this work. Quantitative methods, for example, could have been used to research other aspects of digital constructive visualization tools (e.g., to investigate whether—or not—they support a better understanding of the data). However, as explained in the previous section, these investigations fall out of the scope of this dissertation.

1.6 Contributions

This dissertation advances the area of Constructive Visualization in the digital realm trough the following main contributions:

- ***Design and implementation of a novel constructive visualization tool***
  I designed and implemented iVoLVER, a tool and visual language that allows for the creation of visualizations with varied data source types. The tool implements the principles of Constructive Visualization (simplicity, expressivity, and dynamicity) and includes a set of novel widgets that extract data from different types of digital artifacts.

- ***Empirical evidence on how different types of visualization approaches affect people's design processes***
  Through a two-part observational study I investigated how people's design approach changes when using two different visualization tools, each implementing a different approach to visualization authoring. This investigation showed how the tools influence: (1) the visualization process, (2) decisions on the visualization design, (3) people's feeling of control and authorship, and (4) their willingness to explore alternative designs.

- ***Characterization of the visualization tools design space in terms of agency and atomicity***
  I investigated how to design visualization tools that support a user-driven, transparent design process while enabling efficiency and automation through a series of design workshops that looked at how both visualization experts and novices approach this problem. The findings of this study highlight *agency* and *atomicity* as two dimensions that can guide the design of visualization tools.

1.6.1 Secondary Contributions

In addition to the main contributions mentioned above, this thesis makes the following secondary contributions:

- The design and implementation of a set of novel widgets that extract data from varied types of digital artifacts.
- A conceptualization of the problem of designing more automated constructive visualization tools; and a catalogue of example solutions.

1.7 Thesis Outline

The remainder of this dissertation is organized around seven themed chapters that relate to each other as indicated in Figure 1.2.

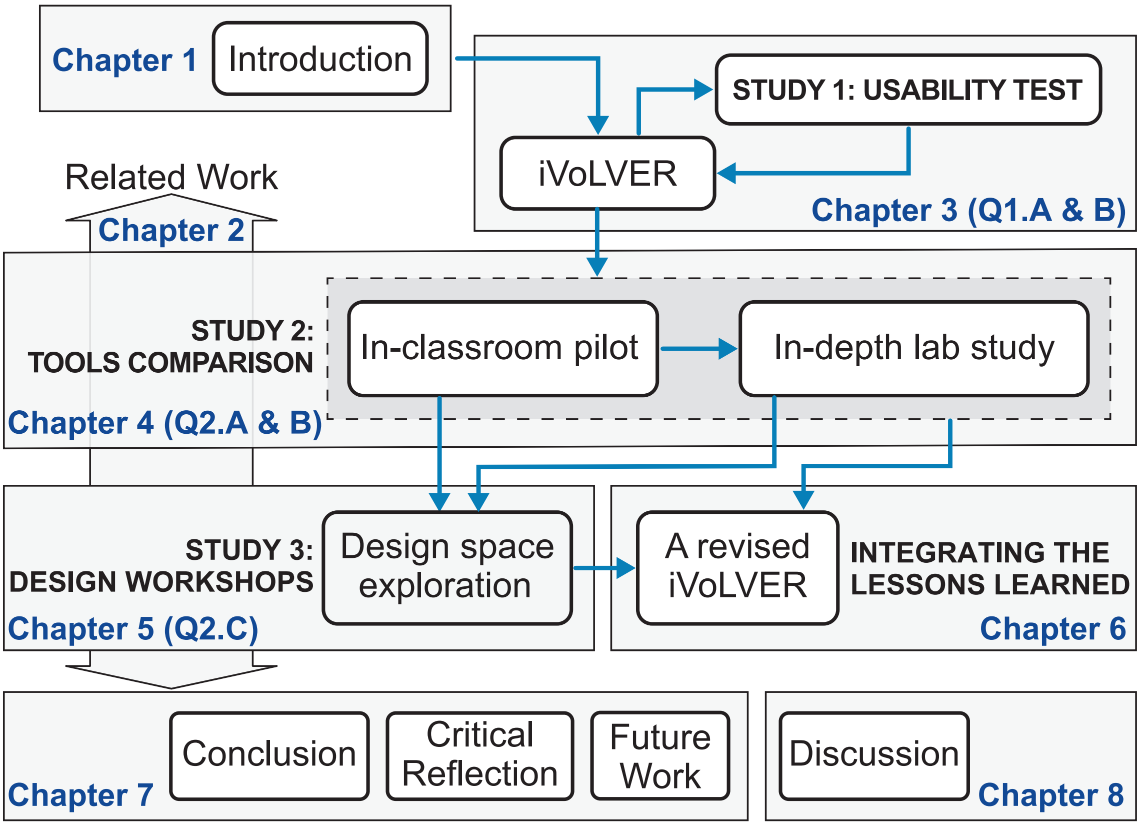


Figure 1.2: Outline of this dissertation. Chapters are represented by the gray blocks. The white rectangles show the main topics/aspects each chapter covers. Chapter 2 spans across the entire dissertation as it includes the related work that is relevant to the overall research.

- ***Chapter 2: Related Work***
  This chapter introduces the most relevant literature to this research. In particular, it presents the theoretical background on information visualization and visual programming languages. It also presents a general description of Constructive Visualization.

- ***Chapter 3: iVoLVER***
  This chapter presents the design and implementation of iVoLVER, the **i**nteractive **V**isual **L**anguage for **V**isualization, **E**xtraction and **R**econstruction, a tool that allows users to create visualizations using a constructive approach.

- ***Chapter 4: Bottom-up versus Top-down Visualization Authoring***
  This chapter defines a continuum spanned between bottom-up and top-down approaches to visualization authoring. It also presents the results of a two-part qualitative study that compared people's visualization processes using two visualization tools located at the ends of this continuum.

- ***Chapter 5: Exploring the Space in Between***
  This chapter investigates how the design space of constructive visualization tools looks like between the two ends of the continuum defined in Chapter 4. The chapter presents the results of a series of design workshops that looked at how both visualization experts and novices design visualization tools that support a user-driven, transparent design process while enabling efficiency and automation.

- ***Chapter 6: A revised iVoLVER***
  This chapter brings together the findings of the experiments of the preceding chapters to propose a revised design of iVoLVER. This design integrates the lessons learned in this research to propose a version of the tool that allows visualization of bigger datasets but still supports a user-driven and transparent design process.

- ***Chapter 7: Discussion***
  This chapter provides an overall discussion of this dissertation. It also presents potential avenues for future research and discusses the limitations of this work.

- ***Chapter 8: Conclusion***
  This chapter holds the conclusion of this thesis and summarizes its main contributions.

Chapter Two
Related Work

The term 'visual language' means different things to different people. To some, it means the objects handled by the language are visual. To others, it means the language itself is visual. To the first group of people, 'visual language' means 'language for processing visual information', or 'visual information processing language'. To the second group of people, 'visual language' means 'language for programming with visual expression', or 'visual programming language.'

— Shi-Kuo Chang [39, p. 1]

As stated in the introduction, most of the research of this dissertation is located at the intersection of Information Visualization (InfoVis) and Visual Programming (VP). In this chapter, I present the general background of these areas to contextualize what is addressed in the remaining of this thesis. The chapter discusses the use of visual representations in the context of visualization and programming, and describes work that is relevant to the overall message of this dissertation. For better readability, research that is more specific to particular parts of this thesis is discussed within the corresponding chapters, in their own "*Background and Related Work*" sections.

2.1 Visual Representations

In his seminal work on semiology of graphics Jacques Bertin wrote: "*Graphic representation constitutes one of the basic sign-systems conceived by the human mind for the purposes of storing, understanding, and communicating essential information. As a 'language' for the eye, graphics benefits from the ubiquitous properties of visual perception.*" [11]. This quote summarizes one of the most relevant features of visual representations: their potential to convey information in recognizable ways without tedious computations [122].

Visual representations are essential in design, science, education, and many other areas. For example, research in educational settings has shown that when students create and interact with visual representations they learn and understand better, as evidenced by several studies (e.g., [63, 65, 139, 183]).

In computer science, visual representations are often used to reinforce human cognition by exploiting the capabilities of the human vision system (e.g., parallel processing, rapid identification of basic visual properties). Within this broader context, several sub-areas deal with visual representations in specific ways. In user interface design, for example, early text-based command-line interfaces have been mostly replaced with other paradigms that heavily rely on graphical elements (such as the WIMP paradigm). Similarly, touch interfaces often support the direct manipulation of visual elements. Among others, computer graphics, visual communication and scientific visualization also deal with visual representations within computational systems. Most of the work discussed in this dissertation is located at the intersection of two sub-areas discussed below: Information Visualization (InfoVis) and Visual Programming (VP).

2.2 Information Visualization

Information Visualization is often motivated as a way to leverage the innate human visual processing capacity for the analysis of data [219]. Keim et al. define InfoVis as "*the communication of abstract data through the use of interactive visual interfaces*" [108]. Creating visualizations can enhance the ability of people to explore, understand, and communicate data by transforming it into an image [215]. Because of this, InfoVis is increasingly having an effect beyond professional scenarios on our personal lives [88, 204], public spaces [16, 85, 210], to tell stories [92, 185], on the way news are communicated [64], and even on the youngest members of the society [6, 134].

There is a current proliferation of tools oriented to simplify the visualization creation process and to make it accessible to non-programmers and non-experts. Making visualization available to a wider audience has many appeals: it can contribute to democratize information [160], it could help people to better understand their own activity and their environments [88], it can promote civic engagement and participation [202], and it could make the public better capable of detecting unethical practices in the use of information graphics [90, 149].

2.2.1 Marks and Visual Properties

Computer based visualization systems "*provide visual representations of datasets designed to help people carry out tasks more effectively*" [143, p. 1]. In a broad sense, the term "visual representations" of this definition refers to the use of geometrical primitives (such as points, lines, squares or circles) to represent data. These elements, called "visual marks" by Bertin [11, 12], are the building blocks of any visualization. For example, in Figure 2.1, the shown scatter plot is composed by a set `points`, the bubble chart is made of `circles`, and the barchart is composed by `rectangles`. Similarly, all the other visualizations of the referred figure are made up of different types of marks such as `lines`, `circular sectors`, or specific `shapes`.

When building a visualization we use the marks' visual properties to encode attributes of a dataset. This process, called *visual mapping* or *visual encoding*, implies establishing correspondences between data attributes and marks' visual properties. Scatter plots encode information by locating points around a specific reference system using their points' horizontal and vertical `positions` (see Figure 2.1). Bubble charts map numeric data to the `area` of their circles and bar charts use the `height` of their rectangles. Other visual properties are `angles`, stroke `thickness`, `width` and `shape`.

Different visual properties have different characteristics and effectiveness. That is, some are better suited to encode specific types of data [44, 45]. Among others, `position`, `length` and `tilt` are better to encode ordered data; while `spatial region`, `color hue`, `motion`, and `shape` are more effective to encode categorical data. The application of these principles is key to create effective visualizations (i.e., visualizations that encode the most important attributes with the most effective channels [143]).

2.2.2 The Current Visualization Tools Landscape

Visualizations are typically built using digital tools and a large variety of these tools exists [34, 70, 121, 135, 150, 200]. The most popular tools for creating visualizations and charts are commercial products such as MS Excel [57] or visual analysis tools such as Tableau [193]. Additionally, online systems such as Gapminder [173] and Many Eyes [214] provide web-based ways to create visualizations, but also share the visualizations and data through the web and support collaboration and comments. These tools facilitate the creation of visualizations without code, but

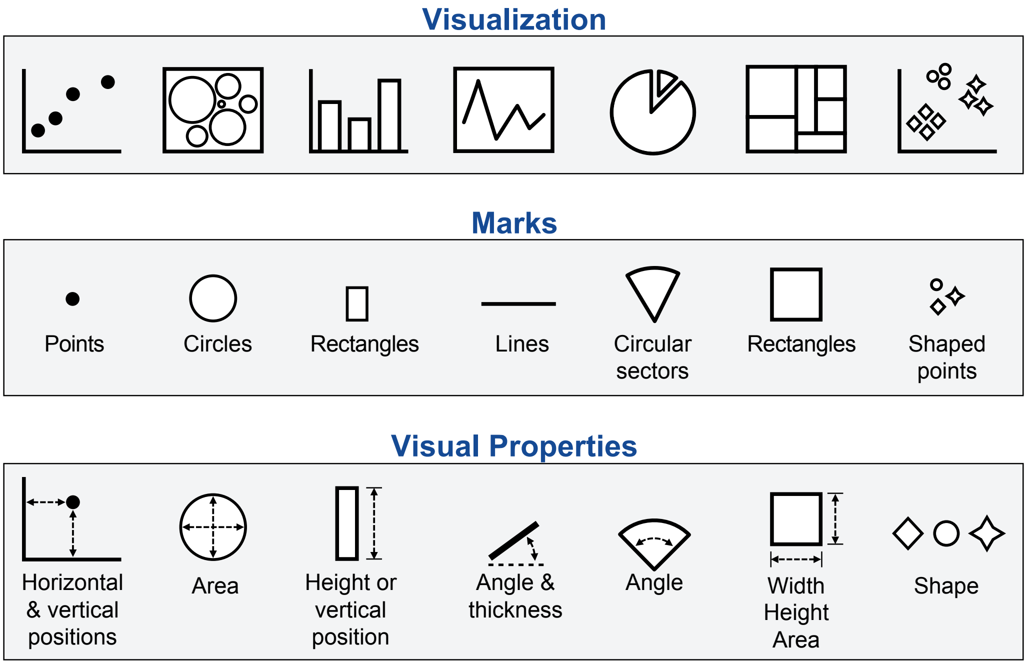


Figure 2.1: Relationships between visualizations, marks and visual properties.

tend to be more rigid and less expressive than programming languages. This is mostly due to the use of specific templates, which limit the choice of possible visualizations to a predefined set of designs and make the data representations somewhat monolithic.

iVisDesigner [165], Lyra [177], and Bret Victor's Drawing Dynamic Visualizations tool (demonstrated in his presentation [212]) enable the construction of more flexible custom visualizations. However, they also include a large number of features, which results in overloaded interfaces with large amounts of hidden data and processing steps that might not be obvious to the user. Chambers and Scaffidi identified this problem in MS Excel [38].

Previous research by Grammel et al. [71] encountered three main barriers that non-experts face when designing visualizations in Tableau [193]. When instructing a tool expert in an experimental setting, non-experts had problems when: (1) choosing the right data, (2) deciding the visual mappings, and (3) interpreting the tool's outputs. These findings highlight the importance of building tools that enable people with no formal training on InfoVis theory to overcome these barriers, which can constitute roadblocks to visualization democratization and visual literacy.

2.2.3 Constructive Visualization

Constructive Visualization (ConstructiveVis) is a visualization authoring paradigm proposed by Huron et al. to empower non-experts in their use of visualization [93, 95]. The paradigm was originally motivated by the democratization of visualization technologies [213], but addresses a set of design challenges that ultimately democratize the visual mapping step of the visualization process. Because of this, ConstructiveVis offers opportunities for populations beyond data experts to not only use tools but also actively engage in the design of their visualizations.

This paradigm to visualization authoring is grounded in educational and learning theories, such as Froebel's *gifts* [198], Papert's constructionism [152], and Piaget's constructivism [3, 157]. This body of research suggests that humans discover the world by manipulating simple objects and that we construct knowledge and meaning from these experiences. From these theories, ConstructiveVis imports the idea of using *tokens* (e.g., Lego blocks) that can be mapped to data and manipulated to compose representations of it. Thus, ConstructiveVis is defined as *"the act of constructing a visualization by assembling blocks, that have previously been assigned a data unit through a mapping."* [95, pp. 436].

ConstructiveVis promotes a visualization authoring process based on three main principles:

- ***Simplicity:*** It does not requires any kind of expertise or special skills (e.g., programming);
- ***Expressivity:*** It allows people to express their own ideas with sufficient freedom; and
- ***Dynamicity:*** If needed, it supports changes in the visualization in response to changes in the data.

Together, these principles constitute an alternative to the way in which most existing tools support the implementation of visualizations. ConstructiveVis: (1) avoids the lack of flexibility and the limits to creativity imposed by template-based tools, (2) overcomes the static nature of the results achieved with freehand sketching, and (3) supports data dynamics without the need of textual programming. The paradigm also overcomes some of the problems that non-experts find when working with more conventional visualization tools. Empirical evidence has shown, for example, that non-expert users did not find the problems observed by Grammel and colleagues [71] when assembling physical tokens to represent data [97].

This dissertation draws from the principles proposed by ConstructiveVis to investigate them in the realm of digital visualization tools. I address this problem by supporting the implementation of visualizations in a visual programming environment. The following sections present the relevant background for this part of my research.

2.3 Visual Programming Languages

A visual programming environment supports the implementation of programs by manipulating graphical objects—rather than specifying them as sequences of written instructions. The syntax and semantics of a visual programming language (VPL) specify what objects can be used to construct programs and what type of spatial arrangements these objects can define.

Although no notation is superior to others in an absolute sense [72] and VPLs are not flawless [20, 146, 220], research suggests that they can be more expressive than textual notations [66]. Burnett et al. [32] list four benefits that VPLs often provide: (1) they reduce the number of concepts needed to program, (2) allow a more direct exploration of the data, (3) make the relationships among the program's components more explicit, and (4) can provide immediate visual feedback. In consequence, VPLs can be beneficial to make programming easier, more reliable, and more accessible to non-programmers.

Some VPLs are particularly good to support the development of computational thinking skills such as Scratch [128, 166, 167] or Mindstorms [151] and, more recently, Google's Blockly library [67]. The syntactic structure of these languages (i.e, the way they use the screen space [123]) is based on blocks that are assembled together to animate interactive visuals. This program construction strategy is also inspired by previous work from Papert [151, 152] and others. Besides block-based languages, other types of visual programming paradigms include spreadsheets, iconic systems, and dataflow environment. Visual Dataflow Programming (VDFP) is particularly relevant for this dissertation and described in detail in the sections below. For detailed descriptions of other types of VPLs, the interested reader is referred to surveys of the general area of Visual Programming (e.g., [23, 31, 48, 144]).

2.4 Visual Programming for Visualization

Arguably, the most powerful and flexible way to create visualizations is by programming them using either general graphical API's (such as OpenGL) or, more effectively, by using specialized visualization languages and APIs such as D3 [24], the InfoVis toolkit [58], Prefuse [81], Processing [164] or APIs for deployment in specific types of environments (e.g., ubicomp environments [8]). However, textual programming requires a significant time and effort to learn and demands a specific kind of environment (e.g., specialized IDEs), which is why a large literature of work in visual programming and visualization tools exist.

The area of Visual Dataflow Programming (VDFP), a subfield of VP that uses "boxes and arrows" to represent data processes and flow, respectively, is particularly relevant for this dissertation as discussed below.

2.4.1 Visual Dataflow Programming

Visual dataflow programming languages (VDPLs) represent programs as a set of connected nodes that process incoming data and exchange their outputs through connections. They have several benefits over other visual programming approaches. VDPLs are particularly suitable to: (1) depict parallel computations and distributed processing tasks [30, 203]; (2) achieve high levels of liveness (i.e., immediate visual feedback [201]); and (3) support live editing while executing. In addition, their visual syntax is arguably closer to the way in which people naturally think when approaching data processing problems which, in turn, facilitates communication with non-programmers and domain novices [9, 104].

Several successful commercial tools have implemented dataflow visual programming environments during the past three decades. Notable examples include Max [50] for audio and video processing, Simulink [131] for modeling and simulation, RapidMiner [163] for the implementation of machine learning programs, and LabView [207] for systems engineering. This visual programming paradigm has also been used in research as a way to facilitate investigation and creation of performance-sensitive data processing algorithms [174]. General surveys on other applications of visual dataflow programming can be found in [83] and [104].

2.4.2 VDFP for Visualization Authoring

VDFPs have also been used in visualization authoring interfaces. GADGET/IV [62] and VANISH [107] provide boxes and arrows programming interfaces as a means to generate visualizations in a separate visual space. A more recent example is the visual IVO editor [132] which makes visualization programming of image-space visualization operations for the GPU more accessible to non-programmers.

Javed and Elmqvist used a VDPL to address the issue that current tools do not provide an adequate representation of the visual analysis process and built an infinite canvas with exploration plates which can be interconnected and annotated in a pen-based environment [103]. Some of the philosophy in this system can be traced to previous work in the DataMeadow system [56], which also uses an infinite canvas but for the analysis of large-scale multivariate data.

This research includes the use of visual dataflow programming to implement iVoLVER, a visualization authoring tool that incorporates constructive principles, as explained in the next chapter.

Chapter Three

iVoLVER

When one can superimpose, juxtapose, transpose, and permute graphic images in ways that lead to groupings and classings, the graphic image passes from the dead image, the 'illustration,' to the living image, the widely accessible research instrument it is now becoming. The graphic is no longer only the 'representation' of a final simplification, it is a point of departure for the discovery of these simplifications and the means for their justification. The graphic has become, by its manageability, an instrument for information processing.

— Jacques Bertin [12, p. 4]

As stated in the introductory Chapter of this dissertation, so far, ConstructiveVis as a paradigm for visualization authoring has only been investigated through the manipulation of physical tokens. In their early investigations, Huron and colleagues studied how non-experts constructed visual representations with tangible tokens [97]. In a latter study [223], they looked at how this type of visualization process compares to the one of a digital tool—MS Excel [57]—for the creation of bar charts. These studies uncover interesting insights on the benefits and limitations of ConstructiveVis and hint at its effectiveness as an alternative to more conventional visualization authoring paradigms.

However, arguably, the vast majority of real-world analytical scenarios involve the use of digital tools and datasets to produce outcomes that are also digital. In consequence, implementing ConstructiveVis exclusively via physical tokens seems unfeasible. This raises the question of how could the principles of this paradigm (simplicity, expressivity, and dynamicity) be integrated in fully digital visualization tools (Q1).

This chapter presents the design and implementation of iVoLVER, the **i**nteractive **V**isual **L**anguage for **V**isualization **E**xtraction and **R**econstruction. iVoLVER is a prototype tool designed to fill the gap in the design space of visualization tools and ConstructiveVis highlighted above. The tool design is driven by the principles of ConstructiveVis and by an additional set of specific goals motivated by making visualization available in a wider set of scenarios (e.g., casual [160] and opportunistic [136, 137]).

As a tool that promotes visualization in "opportunistic" settings, iVoLVER is designed to enable flexible acquisition of many types of data (text, colors, shapes, quantities, dates) from multiple source types (bitmap charts, web pages, photographs, SVGs, CSV files). This way, the tool seeks to support "*data analysis anywhere, anytime, from anything*" [137, p. 234]). To this end, iVoLVER supports, within the same canvas, a constructive process to transform and visualize data, and to build interactive animated visuals.

Aside from the tool, which is web-based and designed for pen and touch, this chapter contributes iVoLVER's visual language and widgets for extraction, transformation, and representation of data. The flexibility and expressive power of the tool is demonstrated through a set of scenarios. Finally, this chapter discusses some of the challenges encountered in the design and development of the tool and how it fits within the current InfoVis tools landscape.

3.1 Background and Related Work

In this dissertation, I propose iVoLVER as the first step to materialize ConstructiveVis in a digital tool. Before describing iVoLVER's design principles, the subsections below first report existing empirical results on ConstructiveVis. This is followed by a motivation to support visualization in opportunistic settings and a summary of the visual programming and visualization tools that form iVoLVER's conceptual base.

3.1.1 Empirical Explorations of ConstructiveVis

The first experimental investigation on how ConstructiveVis can empower people in their use of visualization involved the use of physical tokens (wooden tiles from a learning kit based on Froebel's Kindergarten ideas [130]—see Figure 3.1). This study [97] showed that people with little experience in visualization authoring were able to create and explain their own visualizations in an experimental setting that also involved updating and annotating their constructions. The study revealed

Figure 3.1: Physical tokens used to investigate how people build data representations following a constructive approach. Image taken from [97].

that people generally followed eleven different sub-tasks to achieve three main goals: construction, computation, and storytelling. Participants of this study did not find major problems when designing visual mappings (as opposed to Grammel et al.'s observations with Tableau [71]). Additionally, unlike required by common visualization tools, participants did not need to take early decisions on visualization techniques to begin their data representation process.

A second study involving ConstructiveVis explored how a constructive visualization process compares to the one of a digital tool (MS Excel) for the creation of bar charts. In this work, Wun et al. [223] used the same tangible tokens of [97] and found that the process people follow in these two conditions significantly differs in: (1) the sequences of actions they take, (2) how they distribute and spend the time on different aspects of the InfoVis pipeline, (3) how they separate the tasks of the pipeline, and (4) the flexibility with which they can manipulate visual variables. This study expands the results presented in [97] and sheds light on the differences that a constructive tangible visualization process exhibits in comparison to a more conventional computer-supported authoring process.

Although the studies described above highlight several benefits of ConstructiveVis, none of them explores this from a fully digital perspective (e.g., using a purely digital constructive visualization tool). This limits the generalizability of their findings and their potential implications in the design of new constructive visualization tools for real-world scenarios.

3.1.2 Visualization in Opportunistic Settings

As explained in section 2.2.2 of the Related Work chapter, a lot of effort has been invested in visualization programming languages and APIs (e.g., [24, 81, 164]), tools (e.g., MS Excel, Tableau, ManyEyes [214]), and research prototypes (e.g., [56, 205]). However, many of the existing tools and systems assume that: (1) the data is available in a digital structured format, (2) that the user can access the software when it is needed with the appropriate input and output devices (e.g., keyboard for textual programming), or (3) that the user has programming and debugging expertise. These assumptions make it harder for non-programmers and non-experts to take advantage of InfoVis in "opportunistic" settings (e.g., in situations where textual programming is not possible or when the data is not in the right format). This, in turn, harms the democratization of visualization technologies as it decreases the scenarios in which visualization authoring can take place.

Data might exist in forms that are not easily processed by existing digital tools. Several research and production systems address this problem. ReVision [178] applies computer vision to recognize the type of chart, extracts marks and visual encodings from raster images, and allows the user to remap the data to different visual variables. ReVision builds upon existing work that uses Computer Vision (CV) approaches to extract data from existing graphics, mostly for the purpose of making the quantitative information of existing documents available as searchable semantic data [89, 120]. Harper and Agrawala also provide a toolset that enables remapping of visual variables in existing D3 visualizations [75] to generate alternative graphics. Poco et al. [105, 159] extract and retarget visual mappings from bitmap images of visualizations. Additionally, tools such as WebPlotDigitizer [172] ans Data Thief [208] enable the extraction of data from digital charts.

iVoLVER shares with these systems the goal of allowing users to extract and, in some cases, modify and restyle visualizations. It also allows to derive new data and to combine multiple sources. iVoLVER solves the problem of extracting the data by allowing users to drive the extraction process via simple gestures or interactions. In other words, it is the user who decides what is a mark or an encoding and how to extract the visual variables from an existing graphic. In this sense iVoLVER is closer to the concept of Transmogrification [28], proposed by Brosz et al., where gestures on existing bitmap graphics let the user transform the geometry of existing raster images (e.g., a rose chart into a bar chart). iVoLVER can perform some of these reconstructions, but differs from Transmogrification in that, once the data is extracted from the raster image, it allows calculations and mappings that are not strictly geometrical.

In terms of its ability to collect and combine data from different sources and formats, iVoLVER relates to the Sandbox [222], where pieces of evidence can be manipulated in the context of hypotheses analysis, and to the IdeaMâché curation system [115], which is specifically oriented to support ideation processes, not data representation.

3.1.3 iVoLVER within the current Tools Landscape

iVoLVER shares multiple elements with existing visualization tools such as ExPlates [103] and the DataMeadow system [56], as it supports the construction of visualizations within an infinite canvas via direct manipulation [187]. Like iVisDesigner [165], Lyra [177], and Bret Victor's Drawing Dynamic Visualizations tool, iVoLVER also aims at flexibility. However, it differs from these tools in: (1) its focus on integrating the acquisition of many forms of information, (2) its aim to preserve a visual record of the process as in [103], and (3) its touch and pen-friendly interface.

SketchStory [118], SketchVis [29], and napkinVis [40], are a set of tools designed to support sketching of visualizations in a natural, casual and occasionally social way. As with iVoLVER, a flexible input and versatile visual interface are an important goal of these tools, more so than in Lyra and iVisDesigner. However, iVoLVER provides more computational support and operations (further flexibility), supports a wider variety of data sources, and is based on a more widget-oriented approach rather than on digital ink (thus, it does not rely on ink recognition).

iVoLVER's explicit representation of data processing is inspired by Lark [205], a research prototype that provides an explicit visual representation of the visualization pipeline as interface elements in a tree. iVoLVER applies this approach but at a lower level of abstraction (data point and mark instead of data tables).

Finally, the interface of iVoLVER inherits interaction styles from Pad++ [10] (zooming and panning [47]), and its main interaction paradigm (touch on small objects in a large canvas) relates to several other novel interfaces for music visualization, scatterplots and networks [124, 175, 179].

3.2 Design Goals and Principles

The previous section summarized a long history of tools to create visualizations, some notable examples of tools to transform and adapt existing visualizations, and multiple examples of diagram-based visual languages as a paradigm to specify visualizations. iVoLVER builds upon these works to address the following goals:

- ***G1:*** implementation of a constructive approach to visualization authoring,
- ***G2:*** visualization in opportunistic settings (i.e, in scenarios where data does not exist in database-like documents and has to be extracted from different digital artifacts),
- ***G3:*** flexible exploration of alternative representations (i.e., supporting changes in the visualizations with relative ease), and
- ***G4:*** use by non-programmers and non-experts.

Naturally, the design space of such tools is large. The initial positioning of iVoLVER as a constructive visualization tool can be described in terms of three dimensions:

- ***Atomic vs. abstract:*** Directly inspired by the vision of ConstructiveVis [95], iVoLVER aims at providing concrete interactive visual representations of the smallest atomic elements of data and visualization. This was decided under the assumption that a bottom-up approach to visualization authoring would fit well with the fundamentals of visualization design (Bertin [11, 12] and Cleveland and McGill [44] decompose data representation into atomic building blocks such as marks and elementary graphic encodings). This also supports flexibility (G3) and enables the expressivity principle of ConstructiveVis. This dimension distinguishes iVoLVER from other tools that hide/group operations and data under layers of abstraction through more complex widgets (e.g., ExPlates [103] or Lark's coordination points [205]) or support operations on collections of values (i.e., at the attribute level).
- ***Programmable vs. configurable:*** iVoLVER seeks to achieve a flexible (G3) and expressive visualization process by avoiding the use of templates (unlike tools like MS Excel's charts gallery [57], Gapminder [60], Tableau [193], or Many Eyes [214], where WIMP elements determine the appearance and behavior of the visual representations). As a visual environment, the tool avoids textual programming (G4) and keeps the visualization creation process simple. The key

is to allow any number of elements to be created and to enable their inputs and outputs to be connected in flexible ways. iVoLVER's VDFP approach has been successfully applied before in visualization tools [56, 107].

- ***Integrated vs. specialized:*** One of the main goals when designing and implementing iVoLVER was to create a tool that makes visualization available in a wider range of scenarios by integrating data from multiple sources. The idea was to support settings such as journalists verifying data from several sources or scientists creating alternative visualizations from published graphs (e.g., for discussion). Although it might be possible to take advantage of specialization for specific scenarios, or build multiple tools that address specific needs (e.g., data extraction separate from data manipulation and representation), iVoLVER's design supports all these activities within the same tool, enabling workflows that can quickly alternate among them.

Based on these goals and starting points, the following principles were prioritized when designing the tool. The iVoLVER's user interface (UI) and visual language should, when possible:

- ***DP1–Enhance Atomicity:*** make the core conceptual elements directly accessible to enable an incremental, constructive process in which complex visual constructions result from manipulating simpler ones (G1, G3).
- ***DP2–Visual explicitness:*** make the visualization process and the the interface elements as explicit as possible (G4).
- ***DP3–Avoid textual programming:*** avoid the more complex elements of textual coding (e.g., syntax) to achieve a simple constructive visualization process (G1, G4).
- ***DP4–Avoid menus and hidden operations:*** as in DP3, non-transparent computation and hidden menus might constrain tool use (G1, G4).
- ***DP5–Provide access to data from multiple sources and in multiple forms:*** flexibility and convenience in data input will make the tool more useful when the data does not exist as a structured file, or is not explicitly available to the user (G2).
- ***DP6–Enable interaction through different input modalities:*** To support visual data analysis in a wider set of scenarios (e.g., away from the desk—G2, G4).

3.3 Constructive Visualization in iVoLVER

Together, the DPs of the previous section achieve in iVoLVER's the principles of ConstructiveVis as explained bellow:

- ***Simplicity:*** Constructive visualization is simple "*in that the skills required to build and manipulate the visualizations are akin to kindergarten play*" [97, p. 433]. iVoLVER attempts to achieve this by supporting a visualization process based on the direct manipulation [187] of digital elements within an open canvas. Although the ability to manipulate digital objects is not a skill that we normally learn in kindergarten, it is a minimum required for any digital tool. As a VDFP environment, iVoLVER avoids the need of textual programming (DP3), keeping the visualization construction process simple. DP2 also allows for simplicity as it enables direct access to the elements used to create the visualization (e.g., datasets, marks) within the same space and at all times. Similarly, DP4 does this at the level of the UI (providing easy access to all the tool's features).
- ***Expressivity:*** iVoLVER's atomicity principle (DP1) allows to decompose the visualization process into small units of data and visual representations—the tokens—that users manipulate to incrementally generate visualizations. This direct manipulation of atomic elements achieves a constructive process of high expressivity, as it provides users with freedom and flexibility to customize their visualizations at a very fine-grained level. Additionally, given that this manipulation takes place within an open, infinite canvas, visual marks and other objects can be freely arranged, which further supports expressivity. This is in contrast to other digital tools that restrict the manipulation of the visualization elements to specific spatial arrangements and layouts (e.g., Tableau's "Columns and Rows" system).
- ***Dynamicity:*** The digital nature of iVoLVER achieves a dynamic process in which the visualization can change not only in response to changes in the data but also as a consequence of the user's actions (e.g., when manipulating interactive elements such as sliders). Furthermore, because it is a digital tool, iVoLVER takes the dynamicity principle to a different level. In tangible ConstructiveVis each token of a given representation has an associated unit that is fixed to all the elements of the visualization. This unit must be decided at the beginning of the authoring process and it impacts how

> the overall visualization is perceived. On the contrary, iVoLVER marks afford dynamicity also at the level of the units each one represents. In contrast to their physical counterparts, the properties of digital objects can be modified with relative ease. For example, unlike most physical tokens, the `size` and `shape` properties of digital tokens can be changed to fit the needs of a particular construction.

iVoLVER's design is also motivated by the need to support visualization in opportunistic settings. In this regard, DP5 directly supports analysis of data that does not exist as a database-like document; while DP6 enables analyses with a wider variety of input modalities, enabling visualization beyond the desktop [170].

3.4 The iVoLVER Visual Language

This section describes the iVoLVER tool as well as the basic elements of its visual language. References to the design principles (e.g., DP1) indicate the design motivation.

3.4.1 Canvas and System-Wide Features

The main space of the application is a web-based infinite zoomable canvas or sandbox where all objects are dropped and can be relocated at will by dragging (DP1, 2, 4, 6). Most widgets have two states: compressed and expanded. *Compressed* is their natural state for display, whereas the *expanded* mode exposes their visual properties and allows connection to other objects.

A top toolbar (Figure 3.2.A) enables essential operations (load, save, align, compress, hide, duplicate, and delete objects), and data importing functions: loading bitmap and vector graphics, structured data files; taking camera pictures; and opening web pages (DP5). The right-side palette (Figure 3.2.B) is separated by categories and contains all the types of objects that can be composed in the canvas. The shape of the buttons encode whether these are draggable (circles) or involve switching to a mode (squares). The palette and the toolbar were designed to show all available functionality at a glance. Although these might appear complex (58 icons), and many icons need to be learned, they also make every operation and widget visually explicit (DP2). This design avoids the multi-step interactions required in menus (DP4—particularly important for touch interfaces), and the discoverability problems of gestures.

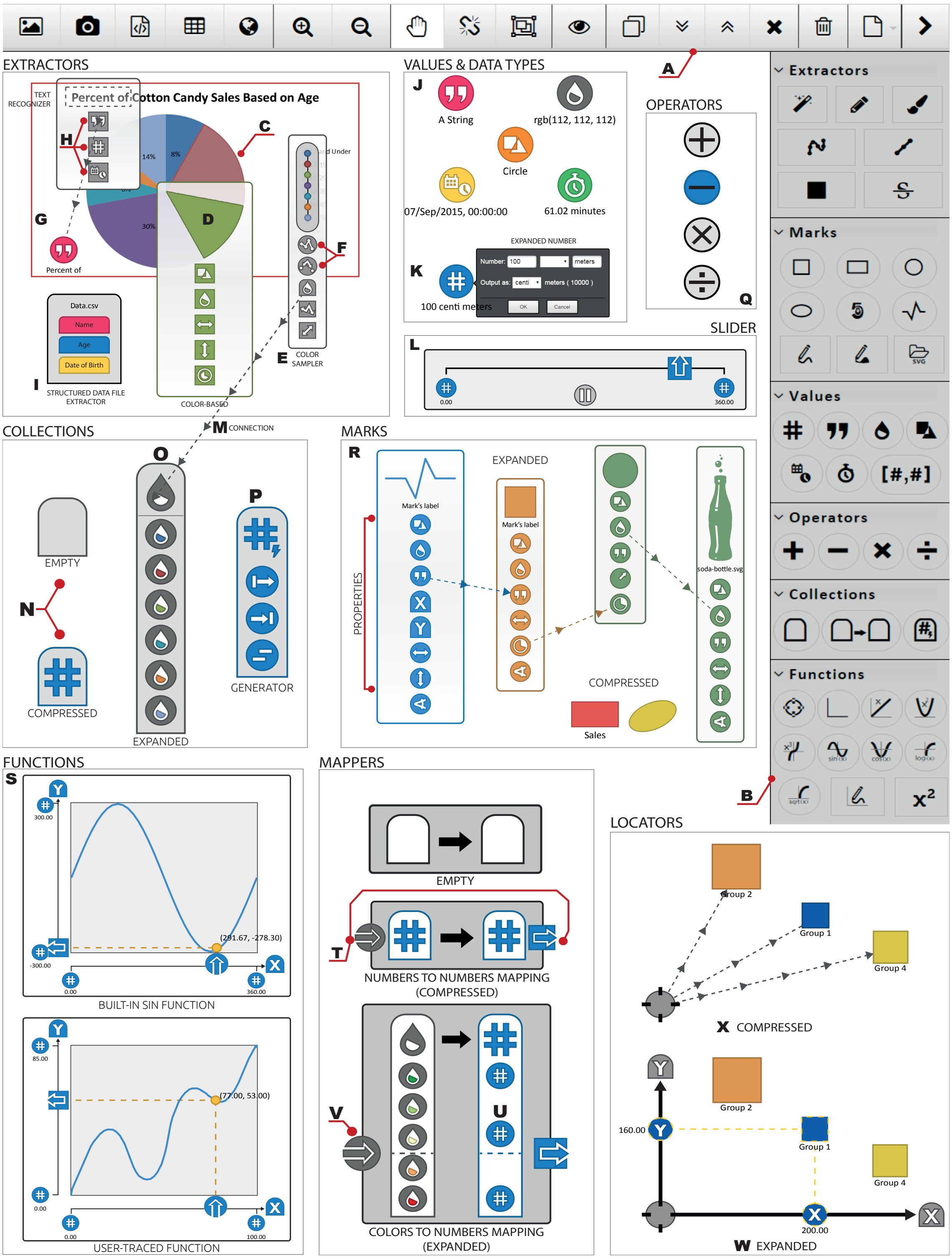


Figure 3.2: iVoLVER interface and elements of its visual language.

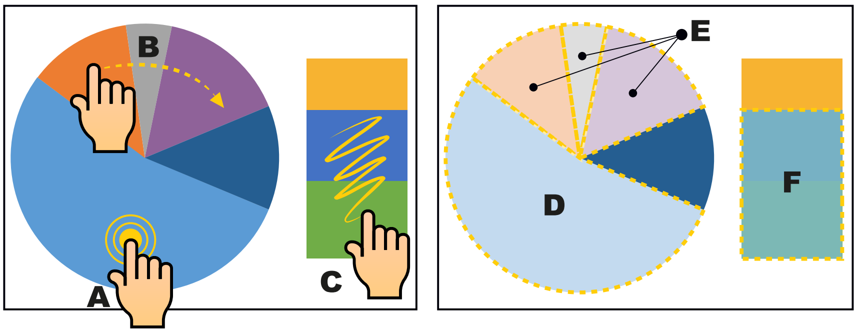


Figure 3.3: Gestures to create color region extractors. A double tap (A) extracts from an homogeneous region (D). A trace (B) creates separate extractors for each of the homogeneous regions underneath (E). A squiggle (C) creates a single extractor (F) from multiple regions.

3.4.2 Extractors

These components extract data from different types of digital artifacts (DP5) and make the extracted data available to other canvas elements:

- ***Color region extractors*** (Figure 3.2.D) can extract size, color and shape from uniformly colored areas within raster images (e.g., a sector of a pie chart, the area of a country in a map). Double tapping on a region of a bitmap image uses a flood fill Computer Vision (CV) algorithm and creates an extractor object from where values can be dragged (Figure 3.3.A). A similar process enables extraction from vector formats (without requiring CV). To facilitate extraction of multiple regions, a trace gesture creates multiple extractors for all objects in the trace (Figure 3.3.B). A squiggle over several areas with different colors creates a single extractor for all areas (Figure 3.3.C). If the source material is not good enough for CV (e.g., low-contrast photo), users can still trace areas with a pen or a finger over the picture or map and use their information in the same way.
- ***Line samplers*** are created by tracing a freehand line on top of an image. The sampler extracts colors of the image below (Figure 3.2.E) at adjustable intervals (Figure 3.2.F) and outputs them as color collection. Line samplers can also output the length of their trace. There is a straight line version as well.
- ***Text recognizers*** appear as scalable and rotatable rectangular shapes that recognise textual information underneath the area that they cover (Figure 3.2.G). Expanding a text extractor exposes the recognised text,

which can be interpreted as a string, a number, or a date with a time stamp (Figure 3.2.H).

- ***Structured data file extractors*** are the canvas embodiment of structured data files (in CSV or JSON format), which take the form of a named collection of collections (Figure 3.2.I) or can appear as a table directly on the canvas.
- ***Web page extraction*** enables web pages to be opened in a floating window on top of the canvas. Dragging elements from a web page (e.g., text, images, tables) into the canvas creates the corresponding object, interpreting it if necessary (e.g., as a string or number).

3.4.3 Values, Types and Data Flow

iVoLVER supports six data types: numbers, strings, colors, date and time stamps, durations, and shapes, which are all represented as atomic canvas objects (DP1, DP2). Values appear as colored circles with a symbol inside; color and symbol identify the type of the value (Figure 3.2.J). They are sized slightly larger than a fingertip to enable touch manipulation (DP6). Values can be dragged out of a value holder element (e.g., a property of an extractor) into a blank section of the canvas or from the palette. Expanding a number value (with a double click or double tap) enables direct editing, scaling its output in orders of magnitude (for numbers), and assigning a unit label (Figure 3.2.K). This feature can be used, for example, to modify the units a given mark property represents. Strings, colors, date and time stamps, and durations similarly support manual assignment of the value.

Reading and writing operations to and from a specific iVoLVER value or a value holder element are displayed via visual connections. The connectors are dashed lines with directional arrows that indicate flow direction (i.e., which value is the origin and which is the destination, reading and writing, etc.—see Figure 3.2.M, DP2, DP3). All links except the active one can be made invisible through a toolbar button. This is convenient when the canvas becomes cluttered.

To enable interactivity and as an alternative to having to edit a value manually to explore a particular range, iVoLVER has a slider (Figure 3.2.L) which outputs a value in real time depending on the interactive position of the handle. The range of values can be established interactively or from another object. Sliders also have a 'play' button that cycles through the range of values in a loop, producing animations.

3.4.4 Collections

Values can be grouped into homogeneous collections. When compressed, collections appear as a toast shape with a symbol that represent the type of objects held (Figure 3.2.N). When expanded, each value in the collection can be accessed individually (Figure 3.2.O). Collections are created by dragging the collection icon from the palette, and are typeless until the first value is dragged onto it. Collections are built by adding iVoLVER values one by one or generated from other objects, including a sequence generator that produces a collection of numbers between two values with a configurable step increment (Figure 3.2.P).

3.4.5 Operators

Values of most types can be combined through the four basic arithmetic operations: addition, subtraction, multiplication and division, which are represented as circles with the operator's symbol in them (Figure 3.2.Q). Operators are sensitive to the type of value of its inputs and produce a corresponding output; for example, subtraction of two numbers produces a number, but subtraction of two time stamps produces a duration.

3.4.6 Visual Marks

Visual marks in iVoLVER are represented through their own visual elements (DP1, DP2). There are seven types of marks: squares, rectangles, circles, ellipses, paths, filled marks, and SVGs (Figure 3.2.R). Marks are created from the palette except for paths and filled marks that can also be drawn on the canvas. SVG marks require the selection of an existing file.

The visual appearance of a mark is controlled by several properties (i.e., visual variables) that can be interactively adjusted or receive incoming connections from other objects. All marks have a label, color, and a geometrical shape. Most marks have some kind of geometrical dimension (width, height, radius, angle or a combination of these). Paths and filled paths also take lists of point coordinates to define their shape. Note that all these properties can be read or written. For example, one can give a mark the color from another mark, or read the collection of coordinates that makes up a path mark.

3.4.7 Relations: Functions and Mappers

The ability to transform data (e.g., converting elevation numeric values into colors) is critical to support a flexible visualization process (G3). iVoLVER provides two mechanisms to transform data: functions and mappers. These define a relation between a collection of inputs and a collection of outputs.

Functions correspond to the mathematical concept of real number univariate functions: a mapping between subsets of the domain of real numbers where each value in the input is related to exactly one value of the output. The function widget is one of the most important widgets in iVoLVER since it indirectly supports DP5 and provides a visually explicit representation of data transformation (DP2) without the need to code (DP3). There are three main ways to create a function: one can drag a predefined function from the palette (e.g., logarithmic, square, sin), draw the desired shape directly on the canvas, or create a correspondence between x and y values from elsewhere in the system (e.g., using two collections of numeric values).

The function visual widget is designed to resemble how functions are depicted in mathematical education (Figure 3.2.S): there is a vertical `Y` axis that represents the output and an `X` axis that represents the input. Four numeric values (visible as blue circles) control the input and output range. A line between the orthogonal axes represents the shape of the function. Input and output ports—which also serve as interactive handles—move along the input and output axes, respectively, to graphically represent the transformation. The ports are connected through perpendicular lines to a small yellow circle that moves along the function line. Each of the axes have two collection elements that provide access to the coordinates that define the function.

Functions are used by connecting any numeric value from the canvas into the input port, which will result in the input port moving to the corresponding position in the `X` axis and the output port to the corresponding output position. Dragging from the output port will make the transformed value available elsewhere in the canvas. Dragging the input port manually will also produce the same effect interactively. Dragging a collection of numeric values into the input port will generate a collection of transformed values in the output.

Mappers allow the specification of relations between values of different types (e.g., numbers, colors, and dates), as opposed to functions which only work with numbers. They are a key element to support the use of data from different sources (G2, DP5) because data is often represented in different ways in the real world. For example, many topographic maps use color to represent height.

The mapper takes the shape of two vertical collections of values where the left one represents the input domain, and the right one the output domain (Figure 3.2.T). An input port to the left of the input collection (Figure 3.2.U) can receive an input of the same type as the input domain, or simply be slided up and down. The output port to the right (Figure 3.2.V) will output a value within the output domain that corresponds to the vertical alignment of the specific input.

The vertical alignment of input and output elements determines how the input and output domains are related. For example, if an input numeric value of 7 is aligned horizontally with a color value red, any inputs of value 7 fed to the mapper will be output as red. In between elements the mapper creates linear interpolations. Elements in the input and output domains can also be adjusted by dragging them (Figure 3.2.W). Finally, connecting a collection of values to the input port (rather than a single value) will result in a collection of outputs.

3.4.8 Locators

The locator is an object that controls the position of marks, which is an important visual variable. When expanded, a locator resembles a set of coordinate axes with handles similar to functions (Figure 3.2.X). In its compressed form it is just a small circle (Figure 3.2.Y).

Dragging from the core circle of the locator to a mark gives control of the mark's position to the locator. The mark's `X` and `Y` position within the Cartesian space defined by the locator provide a point of entry for values from other objects. The locator also provides an anchor to move multiple objects at the same time—once linked to a locator, they keep their position relative to it.

3.4.9 Interactivity and Interaction

iVoLVER supports multi-touch, pen-based and mouse input, consistent with DP6. Earlier design considerations aimed at taking advantage of the different input modalities by changing the input behavior depending on the type of input (e.g., enabling the creation of connections by simultaneously touching start and end ports), and even taking advantage of pen and touch interactions (as in [61]).

Although this could deliver optimized interactions and a better use of the input bandwidth of specific devices, the final design supports the use the tool in a consistent fashion across a range of devices. Interactions were designed to be both consistent and feasible across the three input modalities (e.g., drag, tap/click) with occasional specific adaptations (e.g., buttons to zoom in and out when using pen or mouse input, enforcing a minimum size of objects for touch).

The web interface runs well on mobile devices such as iPads and large phones, also supporting DP6. The gesture design is consistent across widgets: double tap (or click) expands and contracts, drag moves an object or pans the canvas, dwell and drag pulls a data connection to another object. Multi-touch pinching (with two fingers) changes the zoom level in the canvas. The design of many widgets also introduces interactivity at all levels. For example, users can manipulate input ports in functions and mappers in real time, as if they were handles. Interactive sliders also enable interactive visualizations and animations.

3.5 Design Process

The design process was both iterative and user-driven. The initial stages consisted of sketching sessions whose outcomes I discussed with my supervisor, Dr Miguel A. Nacenta. Later in the process I alternated several design and implementation phases. I also demonstrated an early version of the system to an expert (a researcher in globalization and education who needed to make compelling demonstrations of her non-standard map-based data to politicians) whom I also interviewed.

Demonstrations and interview with the researcher took place twice: once at the beginning of the design and once at the midpoint of the development process. These helped to determine and prioritize the features to implement. Several features were added as a consequence, including the ability to import structured files, and elements to position visual marks relative to a coordinate system (i.e., locators).

The design of iVoLVER included an empirical study carried out to detect early usability problems and to inform further refinements of the tool. The execution of this study was part of Sebastien Vandenheste's masters project at the School of Computer Science of the University of St Andrews. The study resulted in changes of the organization and icons of the tool's menus and the copying, aligning, and link hiding features. A detailed description of the study and its outcomes can be found in [211] or at the ACM Digital library, as part of the source materials of the paper that describes iVoLVER [138].

3.6 Implementation

iVoLVER works under a client-server scheme. The client is implemented as a dynamic web page mostly written in JavaScript. The main area of the screen (i.e., the system's sandbox) is implemented on top of a HTML5 canvas element, where most of the rendering process takes place. Since text recognition and computer vision are not yet well supported in browser-based frameworks, iVoLVER's current implementation offloads these tasks to a server program specialized in image processing routines.

Fabric.js [226] is used to handle the 2D graphical context of the HTML5 canvas element while seeking to exploit the processing capabilities of the GPU (if any) at the client side. iVoLVER also takes advantage of regular DOM elements to implement objects such as the panels used to manually modify values (e.g., to configure numbers' magnitudes). The tooltipster [100] jQuery plugin is used for this purpose.

When running on multi-touch displays, the user input is handled through the Hammer.js [180] library in combination with the built-in touch capabilities of Fabric.js. Other libraries used in the client side of the system are: Simplify.js [4] for polyline simplification support, the JSTS Topology Suite [77] for the application of spatial predicates and geomety processing, and the moment.js [41] library for date parsing and manipulation.

On the server side, a computer vision system written in Java executes several image processing and CV routines as they are requested by the iVoLVER's web client. Among other tasks, the server component is responsible for: (1) computing color-based connected components in images through the application of a flood fill algorithm, (2) finding contours in binary and gray-scale images, (3) applying graphical morphological operators (such as erosion and dilation), (4) performing per-pixel operations (e.g., to extract color samples), and (5) recognizing text. These tasks are implemented with the OpenCV library [99]. The text detection and recognition routines are built upon the Tesseract4Java library [216]. The communication between the client and the server is implemented through Java servlets.

Although eventually it will be possible to carry out the server-side operations in the browser, there are several advantages in offloading them to an internal or external component: (1) computation is offloaded as well, which can benefit the use of the application in devices with low computational power such as phones and tablets; (2) with the appropriate definition of a communication interface, offloading results in a separation of concerns and improves maintainability; (3) it can spur the usage of computer vision computations as a service from different provider frameworks or libraries, which could compete and therefore improve on the basic algorithms.

3.7 Scenarios

This section first shows how iVoLVER can be used to create a simple visualization of a small dataset containing structured data. This is followed by five scenarios that show how iVoLVER supports InfoVis in opportunistic settings.

3.7.1 Visualizing a Structured Dataset

In this scenario, Valery uses iVoLVER to create a bar chart representation of a small structured dataset. She first imports the data using a file browsing functionality available in the tool's top toolbar (C in Figure 3.4). The imported dataset appears on the canvas as a table composed by strings and numbers (A in Figure 3.4). Each of these values are connectable canvas objects. Valery then drags five rectangular marks onto the canvas from the tool's right palette (B in Figure 3.4).

With the first mark expanded (after double-clicking it), Valery connects—via drag-and-drop gestures—values from the table to the mark's visual properties. For example, the string `Emily` is connected to the mark's `label` property, and the number 45 to the `height`. These connections define specific mappings between data and visual properties. On performing a new connection the visual appearance of the mark involved changes accordingly.

Valery then repeats the mapping process for the remaining four marks that compose her visualization. After mapping all the data she arranges the marks using iVoLVER's alignment and distributing tool. This last step results in the bar chart of Figure 3.4.

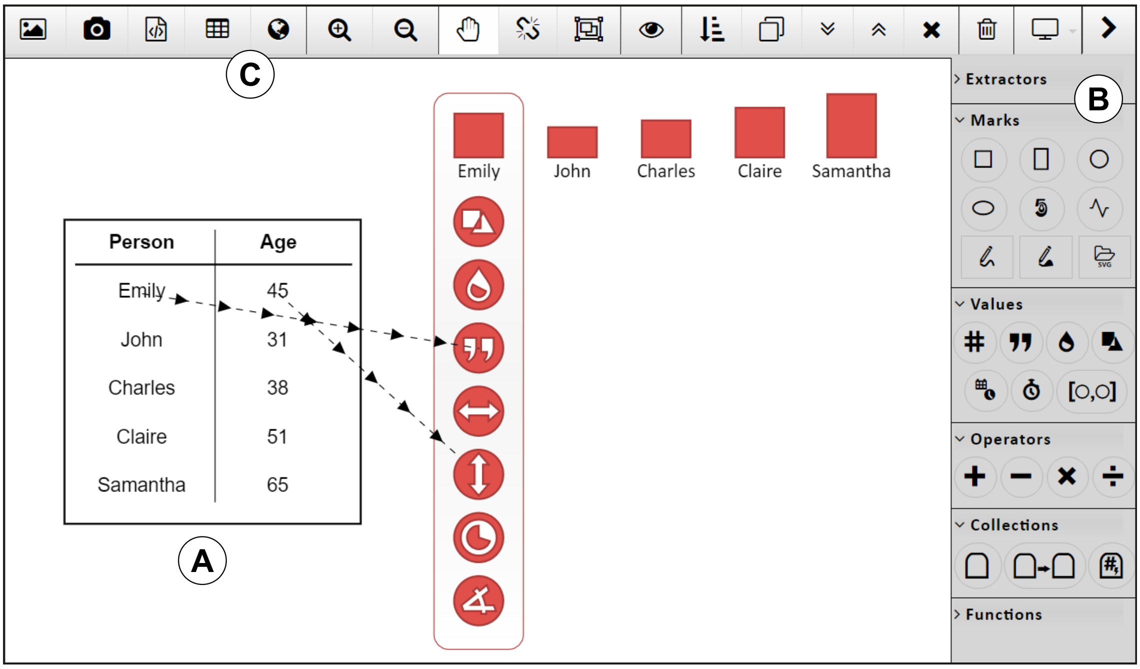


Figure 3.4: iVoLVER used to create a bar chart representation of a small dataset.

3.7.2 Refactoring an Existing visualization

Susan is a professor at the fictional University of La Libertad, working with her student, Daniel, on a paper on world food production. When Daniel brings her the latest version of their paper Prof. Susan is dismayed at seeing a pie-chart (see Figure 3.5.A), which she has learned is a poor way to represent data due to humans' inability to judge angles [44, 45, 80]. Instead of tearing the paper apart and sending Daniel back to his desk to send her the data or re-do the image, she decides to take a more didactic approach.

Prof. Susan captures the image from the paper with her tablet's camera and extracts each of the pie sector areas, plugging them into the length of bars (B). This only took a few seconds, but it is enough to convince Daniel that the barchart would be a better representation. Daniel then suggests that, as he has read recently in a statistical paper, perhaps a logarithmic transformation of the data would provide a better view. Prof. Susan is skeptical, but indulges Daniel's suggestion and applies a logarithmic function (C), plugging the transformed results into bars again (D). Very quickly they both agree that the logarithm was not useful.

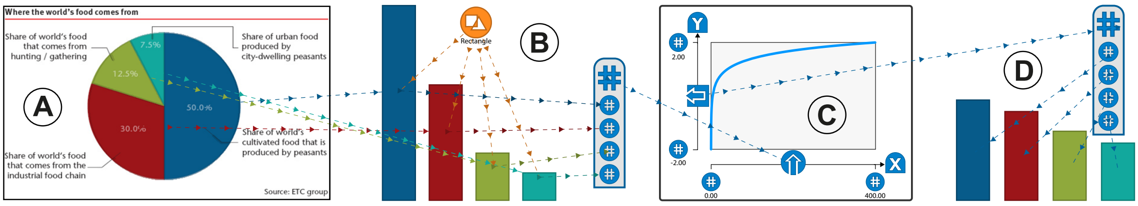


Figure 3.5: A) Data is extracted from a pie chart, B) converted into bars, and C) transformed into D) a log-scale bar-chart (see Motivating Scenario).

3.7.3 Verifying Infographics (and Correcting Them)

Figure 3.6 illustrates how iVoLVER can be used to verify (and correct) mistakes (or manipulations) in existing infographics. This scenario uses the example of a TV line chart shown in the National Spanish Television in January 2015 [127]. The graphic showed a line representation of unemployment that is incongruous with the numbers displayed and with a non-zero baseline. This representation exaggerates and misrepresents unemployment decrease in the last two years, which led to many critics of the former government to claim bias through social networks.

With iVoLVER one can trace the function over the displayed line and adjust the range to correspond to the numbers shown in the chart. A quick manipulation of the input of the function shows that the numbers provided are inconsistent with the shape of the chart (by hundreds of thousands of people). It is possible to generate an alternative representation with four rectangular marks that correspond to the numbers provided, this time with a zero-based vertical axis and the right dimensions. It looks dramatically different.

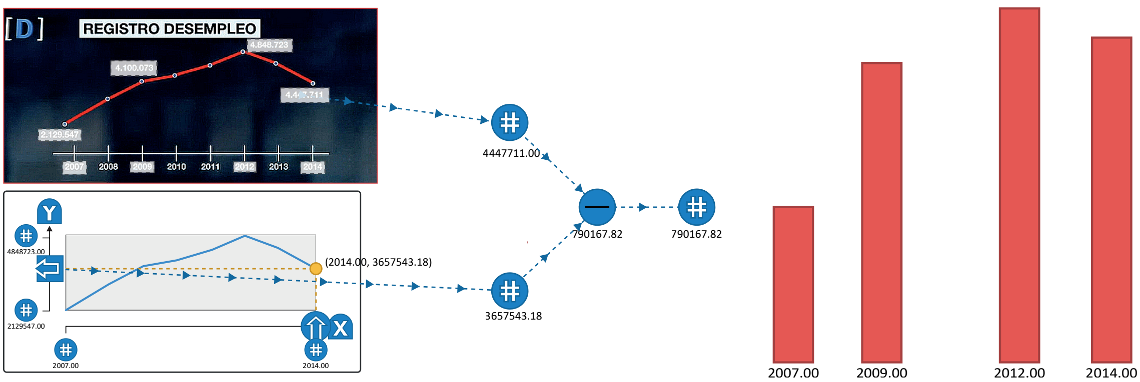


Figure 3.6: Verifying infographics scenario. See 3.7.3 section for a description.

This scenario shows how non-programmers trained in the use of iVoLVER (perhaps even when in front of the TV, with a tablet) could analyze graphics and generate alternative representations that are more understandable to them or less biased. Simultaneously, journalists can use the same procedure to verify their sources and to double check that the graphics that they produce are correct and fair.

3.7.4 Climbing Everest

This scenario, depicted in Figure 3.7, involves the creation of route altitude profiles from raster image data. Two images are imported. One contains a depiction of the North Col-North Ridge and the South Col routes to reach the summit of Mount Everest; the other is a topographic map of the area. The freehand sampler tool is used to create two color samplers that correspond to each of the routes. These traces can be created directly in the second figure, or created in the first figure and then moved onto the second (the second option works in this case as both representations coincide spatially, but it might not be the case in other scenarios).

The colors and the altitude are extracted from the second image's legend to create a mapping between color and altitudes. This mapper can now translate any color or series of colors into a numerical altitude. The altitudes from the route samplers are then fed into the mapper whose output is used to display the ascension as an interactive function. The width of the function widgets are adjusted to scale by extracting the route length from the trace. The result is a set of two comparable profiles.

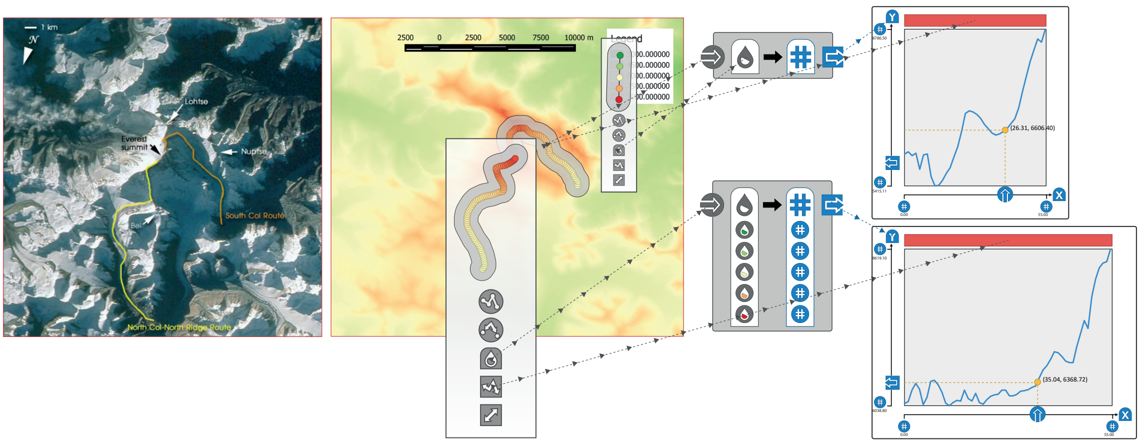


Figure 3.7: Climbing the Everest scenario. See 3.7.4 section for a description.

This scenario exemplifies the creation of a sophisticated visualization optimized for a particular purpose, from data which is spatial and not trivial to extract. Besides the expressiveness of the visual language, this visualization also highlights the embedded interactivity: one can interactively query the profiles, adjust routes or create new ones, adjust the sampling, and any of the extracted or manipulated data can be further fed into new processing or visuals.

3.7.5 Personal Visualization of a Trip

Isabel, who recently learned how to use iVoLVER, just flew from Brest (France) to Perugia (Italy) to visit her family. She decides to visualize the trip in a fun way to her relatives. She uses any map of the area as a base and pulls the temporal data (flight departures and arrivals) straight from her webmail inside the tool (Figure 3.8). She does not need the system to have any geographical knowledge because she can easily select the trajectory. Mappers can transform dates into points of the map, interpolating the points in between. A date slider feeds the mappers to generate coordinates which, in turn, control the position of an SVG mark on the canvas through a locator. The resulting animation is proportional to the actual time scale of the trip. Finally, she can annotate the trip with her tweets because these are also time-indexed.

The scenario demonstrates pulling data from different sources to construct something new and fairly sophisticated in behavior without the need of any textual programming. The result is fun and animated.

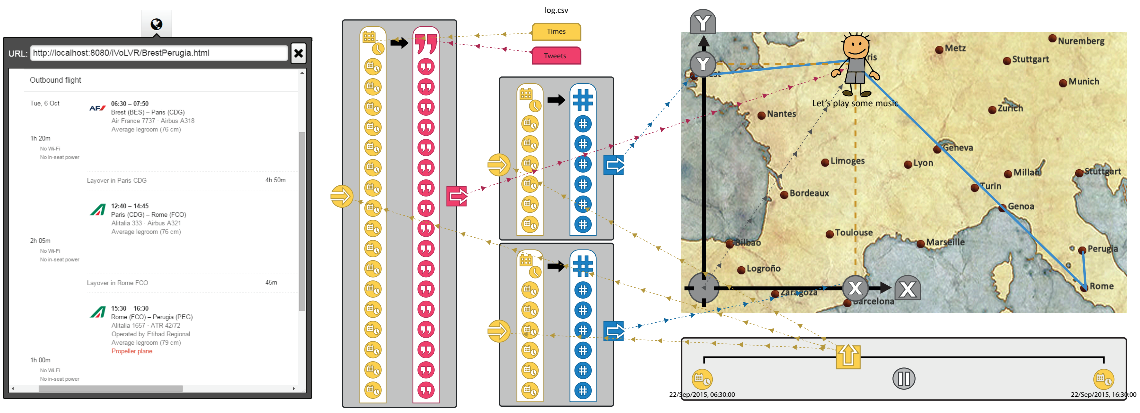


Figure 3.8: Personal visualization of a trip scenario. See 3.7.5 section for a description.

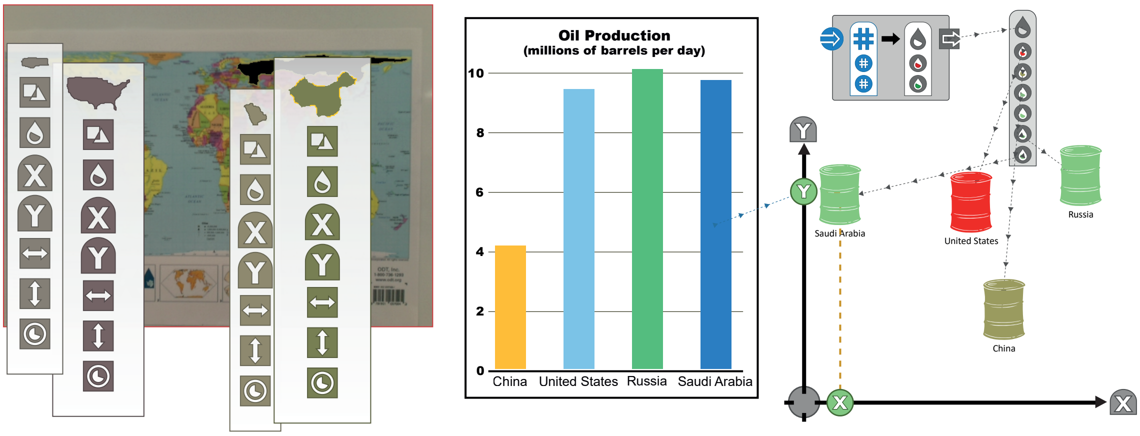


Figure 3.9: Oil production and consumption scenario. See description in 3.7.6 section.

3.7.6 Oil Production and Consumption

This scenario displays the multiple flexible data importing mechanisms of iVoLVER and demonstrates how the different collected data can be combined to build a relatively standard chart of multivariate data (Figure 3.9). The scenario uses the oil production in the top four producing countries as a motivation. First, the photograph feature of the tool is used to import a map that was on a poster. Color extractors pull approximate country areas from the map (with squiggles). Extractors on an SVG bring in the main production data and the labels. The consumption data comes from a table from the web. A locator is used to map the `X` coordinate to country area, the `Y` coordinate to oil production, and a number-to-color mapper translates oil consumption into color.

The scenario showcases how the tool can support a workflow where importing and capturing data can be better integrated and interleaved with the design of representations. Additionally, the resulting visualization is also a visual record of the representation choices and the data sources which can be useful in explaining the chart itself.

3.8 Discussion

iVoLVER's design seeks to support a constructive approach to visualization authoring by implementing the three principles of ConstructiveVis (simplicity, flexibility, and dynamicity). The tool also aims at supporting visualization in opportunistic scenarios, by including in its visual language widgets to extract data from several types of digital artifacts (e.g., photographs, raster and vector graphics, web pages and text). iVoLVER thus can be characterized by the combination of four main features:

- It takes a constructive approach to visualization authoring;
- It provides user-driven data extraction from non-structured data sources;
- It does not require textual programming; and
- It enables analysis away from the mouse and keyboard through a touch, pen or mouse-based interface.

These elements aim to empower non-experts in their use of InfoVis and enable analysis in an extended range of possible situations. A small set of examples of use is demonstrated through the scenarios from section 3.7.

As a point exploration of a large design space, building iVoLVER has also made apparent the impact of some of its design principles. The following subsections identify the challenges found during the design process and reflect on the advantages and disadvantages of the followed approach. It is important to note, however, that much of what is discussed below poses additional questions rather than providing clear answers. These questions deserve further analytical and empirical study and some are, in fact, addressed in subsequent chapters of this dissertation.

3.8.1 Power and Simplicity

iVoLVER's radical atomic approach (by which it represents very low-level data such as single numeric values—the atoms—through separate interactive visual elements) aims at achieving a flexible, expressive tool in which a few types of building blocks could be combined with each other in a multiplicity of ways. This approach is directly inherited from the ConstructiveVis paradigm and it relates to the concept of emergence, whereby an aggregation of entities arises through interactions among smaller or simpler entities [1]. In the case of ConstructiveVis (and, thus, in iVoLVER), the visualization arises as digital tokens are put together to represent data.

iVoLVER's atomic approach arguably resulted in a flexible interface, especially when combined with the infinite canvas approach. However, the power-simplicity trade-off is not really circumvented and complexity creeps up in several ways. For example, the number of basic element types included in the design is not small.

3.8.2 The Scalability Problem

The atomic approach of the tool also means that, even for moderately simple visualizations, it may be necessary to create and interconnect a substantial number of canvas elements. Sophisticated emergent functionality and visualizations are possible but the time, effort, and number of repetitive interaction steps required to put those together increase significantly with the number of elements involved in the construction. Consider, for example, that constructing the bar chart of the scenario presented in subsection 3.7.1 (see Figure 3.4) requires: (1) five drag-and-drop operations to bring the rectangular marks into the canvas, (2) ten connecting operations between values of the dataset and the marks—five connections to the marks' `height` visual property and five to the `label`, and (3) several alignment and layout-related adjustments. Additional refinements to this visualization (e.g., representing each person of the dataset with a different color, or including additional data points) would require even more effort.

I refer to the issue described above as the "*scalability problem*" of ConstructiveVis. In the context of computer performance, scalability "*connotes the ability of a system to accommodate an increasing number of elements or objects, to process growing volumes of work gracefully, and/or to be susceptible to enlargement.*" [22, p. 195]. The implementation of iVoLVER made evident that the visualization process that takes place on digital constructive tools might not *scale* well to larger datasets. That is, the number of interaction steps required to achieve the visualization grows excessively with the amount of data to be represented (number of records and attributes involved). This translates into a significant amount of required time and effort, which could ultimately threaten the utility of the tool in many situations that rely on quick, sketch-like creation of visualizations. Complexity and scalability have been identified as fundamental challenges in existing diagrammatic visual tools (e.g., by Javed and Elmqvist [103]).

Creating visualizations with larger datasets in iVoLVER is tedious also because the tool does not easily allow abstraction (e.g., generalization of the operation on a concrete element to a group or to a group of groups), as enforced by its visual explicitness (DP3) and avoidance of hidden processes (DP5) principles. Chapter 5 investigates strategies to overcome ConstructiveVis' scalability problem.

Some of these approaches are then applied in Chapter 6 to support in iVoLVER the visualization of larger datasets by introducing in the tool higher levels of abstraction in ways that do not obscure or hide the operations that take place in its canvas.

3.8.3 Learnability and Simplicity

Although visual explicitness (DP3), avoiding hidden processes (DP5) and, more generally, avoiding abstraction might be ways to make interfaces easier to learn, iVoLVER is definitely not walk-up-and-use. The learnability challenges from the tool stem, at least partially, from the relatively large number of elements included in its visual language. The multiple types of marks, data types, operations, functions and settings of the interface must all be learned. This might result in a steep learning curve and harm the simplicity constructive principle that the tool pursues.

It should also be highlighted that a visual syntax like iVoLVER's is still a syntax. Unlike physical tokens, the digital nature of the tool and its visual language introduce semantic challenges faced by any programming environment. These challenges do not exist in iVoLVER's tangible counterparts (e.g., the physical tiles used by Huron et al. in [97] and Wun et al. in [223]). In tangible ConstructiveVis, the meaning and function of each token are decided by the designer on-the-fly, while manipulating the tokens. Additionally, the meaning assigned to a particular token can differ from the one assigned to others, or it could even change at any point during the visualization construction process.

Conversely, iVoLVER imposes several predefined constraints on how the objects of its visual language can be used and how they can (and cannot) interact with each other. The tool thus includes a set of allowed operations and interactions on the objects that users manipulate to produce visualizations with it. These constraints are an inherent part of iVoLVER's underlying visual language and could impose learnability challenges that might be equivalent to those of textual programming. Ultimately, this could constitute a thread to the simplicity of the visualization process that iVoLVER seeks to promote.

3.8.4 Integration and Interactivity

One of the main values of iVoLVER comes from being able to extract data flexibly while being able to transform, combine, and represent this data immediately.

Swapping seamlessly between all these activities enables a simple visualization construction process. All the activities take place through the same canvas, which reduces the need to compartmentalize the screen real estate or provide multiple screens or pages, However, this might result in a significant amount of panning and zooming operations, which introduces the need of more effort from the user.

The integration of interactive control of data, representation and parameters within the canvas itself (for example, as interactive sliders) was motivated by the dynamicity principle of ConstructiveVis. The resulting interface is interesting in that the data and its manipulation are integrated within the same space and linked visually.

3.8.5 Limitations and Technical Constraints

iVoLVER does not provide universal coverage of all possible InfoVis tasks or representations. Specifically, it does not provide features for set-level data interaction (e.g., selection, navigation, annotation) and is not as expressive as textual visualization programming languages or other recent tools such as Lyra [177]. For example, there is no simple way to produce force-directed layouts, trees, and two- and three-dimensional scalar fields.

Data extraction from images and text recognition features are also limited by the graphic analysis algorithms. Users have no control on the parameters that guide the image processing routines (e.g., the color similarity thresholds used in the flood fill algorithm). This simplifies the extraction process and avoids complex configuration, but might also impact the user's experience if unexpected results arise.

3.9 Conclusion

This chapter introduced iVoLVER, a digital visualization tool whose design incorporates the principles of ConstructiveVis (simplicity, flexibility, and dynamicity). iVoLVER also supports InfoVis in opportunistic settings, by providing tools to extract data from different types of digital artifacts (raster images, pictures, SVG charts) and visualization away from the desk.

iVoLVER makes the data processing steps—and the intermediate transformations performed on it—visually explicit. This is achieved through a visual language that provides graphical components to extract, manipulate and process both structured and non-structured data, and to represent it through marks (i.e., digital tokens).

The tool aims to support a constructive visualization process in a wider set of situations, such as when the data is not available in a convenient digital form or when the data comes from multiple sources. The flexibility and expressiveness of the tool was illustrated with a set of scenarios. Finally, this chapter provided a discussion that highlights the challenges and lessons learned from iVoLVER's design choices. Next chapter investigates how these choices impact the visualization authoring process of non-experts.

Chapter Four

Bottom-up versus Top-down Visualization Authoring

The art and practice of visualizing data is becoming ever more important in bridging the human-computer gap to mediate analytical insight in a meaningful way.

— Edd Dumbill [54, p. 15]

The previous chapter introduced iVoLVER as a digital tool that supports a constructive approach to visualization authoring. Besides opportunistic scenarios that might require data extraction, the tool can be used to visualize structured data, as demonstrated in the "Visualizing a Structured Dataset" scenario of subsection 3.7.1. This makes iVoLVER comparable—in terms of its goal and potential outcomes—to other existing visualization tools (e.g., Tableau Desktop).

This chapter investigates how the visualization process of structured data takes place in iVoLVER and how it compares to more conventional visualization tools (Q2). The chapter focuses on how iVoLVER's constructive approach to visualization authoring influences the process that non-experts experience when creating visualizations with it. As a result, it uncovers the consequences of iVoLVER's design principles, identifying the benefits (Q2.A) and limitations (Q2.B) that these principles bring into the visualization process of non-experts.

This chapter explores the questions above by comparing the use of iVoLVER with Tableau Desktop, a more conventional tool that automates several steps of the visualization process. After discussing relevant related work, the chapter describes the fundamental differences between iVoLVER—as a digital constructive visualization tool—and more predominant visualization authoring strategies (e.g., template-based, automated design). It then discusses these differences as part of a continuum spanning between bottom-up and top-down approaches to visualization authoring.

The chapter then presents a two-part qualitative study that investigates how iVoLVER and Tableau Desktop, as tools that implement opposite approaches to visualization construction influence: (1) non-experts' approaches to the visualization process, (2) the differences in the visualizations they produce, (3) their design choices, (4) the low-level visual mappings they produce, and (5) their justifications of these approaches.

4.1 Background and Related Work

The subsections bellow discuss the background specific to the content discussed in this chapter. In particular, I review general theory on the visualization design process and describe in detail existing visualization tool types.

4.1.1 The Visualization Design Process

Several authors have discussed the different stages and key elements of visualization design (e.g., [15, 42, 53, 169, 197]). Card et al. [35] describe the process of creating visual data representations as a set of four steps: (1) data analysis, (2) filtering, (3) mapping, and (4) rendering. Step 3—the creation of mappings between data attributes and visual properties—is one of the most important and well researched elements of visualization design. Much of the InfoVis research of the last 50 years focuses on the perceptual properties of visual elements [45] and the fit between the data types and the marks and visual properties to represent them (e.g., [11, 12, 143, 218]).

This chapter investigates the types of mappings that people come up with when designing their own visualizations. This relates to recent research on how people sketch data representations. Walny et al. [217] describe a "representation continuum" that shows a wide range of levels of abstraction when people sketch up representations of small datasets by hand, from *countable* (concrete and numeric) to *pictorial* (abstract) representations.

4.1.2 Visualization Tool Types

Many visualizations tools allow to produce similar outcomes with different interfaces. Based on Grammel et al.'s categorization [70], three types are most relevant to the work discussed in this chapter: template editors, shelf configuration tools, and visual dataflow programming tools:

- ***Template editors:*** According to Grammel et al., in tools of this type *"the user selects some data and then picks a pre-defined visual structure in which to represent it. The distinguishing criteria of this approach are the separation between the initial visualization selection steps and the refinement of the selected visualization"* [70]. Victor [212] groups template editors under the concept of "using tools", in reference to the predefined solutions provided by these tools that people use to represent data. Examples of template-based tools include Many Eyes [213] and MS Excel [57]. Tableau Desktop's "Show me" pane [126] is a template feature within what is mostly a shelf configuration tool (see below).
- ***Shelf configuration:*** These tools allow the specification of mappings through simple GUI operations on collections of data attributes and visual properties that are presented visually in the tool's GUI. Customization is typically based on cursor operations (e.g., drag-and-drop) or menues and dialog boxes and requires additional steps. Tableau Desktop and Lyra [177] (a more recent research prototype) fall within this category.
- ***Visual Dataflow Programming:*** As explained in section 2.4.2 of the Related Work chapter, tools of this type enable designers to create visualizations by connecting graphical elements. The connected components define the dataflow and the characteristics of the visualization. This approach has a long history (e.g., [2, 125]) but is also present in more recent tools such as DataMeadow [56] and ExPlates [103]. As a VDFP environment, iVoLVER falls in this category.

4.2 Bottom-up and Top-down Visualization Authoring

Besides Grammel et al.'s categorization [70], I propose in this chapter to distinguish between more high-level contrasting approaches to visualization authoring: bottom-up and top-down. These two strategies define a continuum of visualization authoring approaches:

- ***Bottom-up:*** In tools that implement this approach one starts at a low level of abstraction, by manipulating individual data values and visual properties to incrementally generate the visualization. ConstructiveVis fits this description, as users create visualizations by assembling blocks, each of which represent a small unit of data [95]. As a constructive visualization tool, iVoLVER also features this approach: users explicitly establish mappings between individual data points and the graphical elements that represent them, and the final visualization emerges as the result of many small-scale decisions and manipulations.
- ***Top-down:*** In tools that apply this approach to visualization authoring decisions about the visual representation of data happen at an attribute level. Template-based and shelf configuration tools inherently follow this approach—they focus on the overall mapping between attributes and visuals first. Customizations can be applied only after the initial attribute-level mappings are created. Data mappings performed in this way typically include all data points of an attribute, or even the entire dataset. The manipulation of the visual representation at an individual data point level is generally not possible (e.g., in Tableau visual mapping actions include all data points associated to a single attribute).

The distinction of these two opposite approaches to visualization authoring is relevant to this dissertation for two main reasons. First, top-down visualization is arguably the most common strategy within the current visualization tools landscape (e.g., most tools operate at the attribute level). Second, bottom-up visualization authoring takes place almost naturally with ConstructiveVis. Huron et al. identified that the most common procedure that people follow to create visualizations when using a constructive tool *"consisted of starting from a single data case, then building progressively towards defining higher level structures"* [97, p. 2109].

Different types of tools and approaches to visualization authoring can potentially lead to different outcomes and design processes. This is relevant for designers of authoring tools. In addition, effective visualizations are achieved by applying design principles (i.e., what visual properties are better suited to encode specific attribute types [44, 45, 143]). Non-experts are usually exposed to this knowledge through the digital tools they use to create visualizations. People therefore learn as they experiment with an interface that might not have been designed for supporting an understanding of effective visualization design. To better support non-experts to leverage the power of visualization it is important to understand how they go about creating visualizations and how the tools themselves affect this process.

The reminder of this chapter explores this question from an experimental perspective, by comparing the use of iVoLVER with Tableau Desktop as representatives of the ends of the bottom-up versus top-down continuum of approaches to visualization authoring. The following subsection first explains these tools' main differences and similarities.

4.3 iVoLVER & Tableau Desktop: Differences and Similarities

As mentioned earlier, shelf configuration and template-based tools (as defined in [70, 212]) are located at the top-down end of the continuum of the previous subsection. Tableau Desktop [193], a current leader in the visualization tool market, is a shelf configuration tool and therefore a representative of the top-down approach.

Tableau's main interface is divided into rectangular panes. Once the data is loaded the user can see a pane containing the different data attributes (A in Figure 4.1). Dragging and dropping attributes from this data pane onto the "Columns and Rows" shelves (B in Figure 4.1) triggers changes in a separate visualization pane

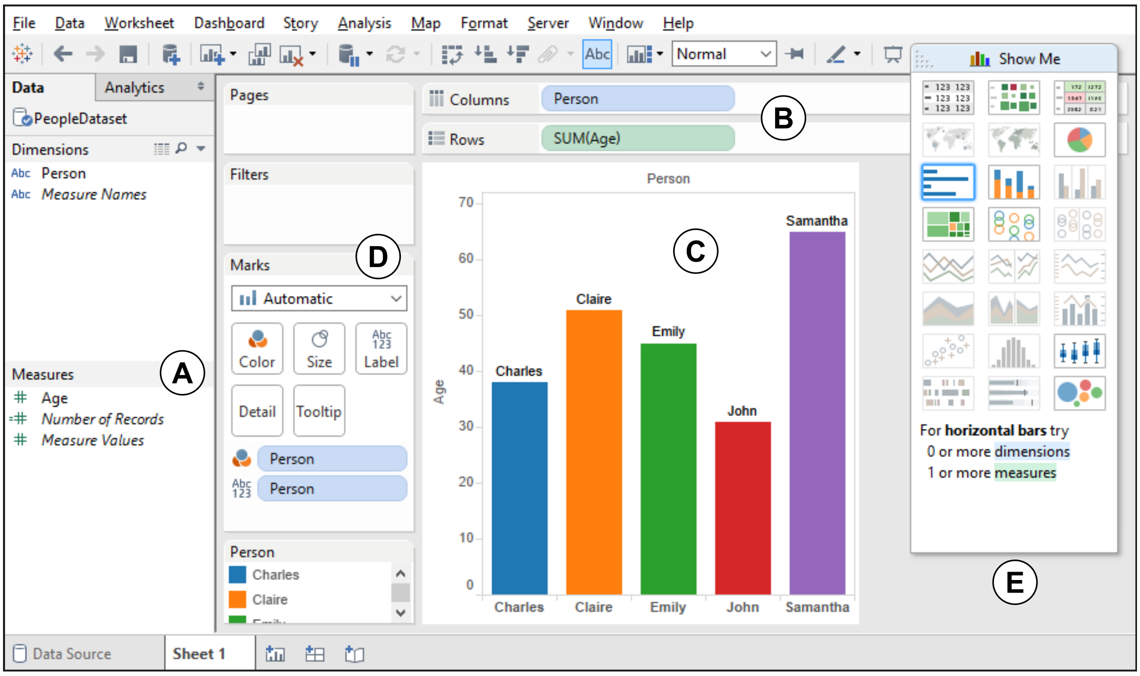


Figure 4.1: Tableau Desktop Interface.

(C in Figure 4.1) that displays a default data representation selected by the tool according to the dragged data attributes. An additional pane enables adjustments of the marks' visual properties by dropping data attributes to other areas of the interface (D in Figure 4.1). A further pane, "Show-me" [126] (E in Figure 4.1), provides a gallery with several visualization templates (e.g., choropleth map, area charts). The templates that are not available with the current attribute selection are automatically grayed out.

Tableau Desktop and iVoLVER are both digital tools and share the use of marks (as defined by Bertin [11, 12]) and visual properties [44, 45] as central concepts in visualization design. Additionally, none of these tools requires textual programming (for simple visualization tasks) and both support dragging and dropping interactions. However, the two fundamentally differ in their underlying design approach, which results in important differences, as discussed below.

- ***Data Atomicity***

  The main activity of designing visualizations in Tableau Desktop consists of the assignment (through drag-and-drop), of a data attribute to a visual property or spatial dimension (row or column). A key point here is that Tableau users do not have the option to manipulate individual data points, but instead deal with the full range of data by attribute. Interactions happen at an attribute-level, including all the corresponding values in the dataset. This enables fast construction of visualizations.

  In contrast, iVoLVER operates at the level of individual data values, a lower level of data abstraction. With iVoLVER users are responsible for defining what is a data value, how it is mapped to a specific visual property, and how it is to be interpreted (e.g., as categorical or numerical data).

- ***Operation Repetition***

  In Tableau, a single action typically affects all elements of the visualization regardless of whether the dataset contains a handful or millions of rows. Thus, Tableau typically does not require repeated interaction steps. In contrast, iVoLVER's paradigm of bottom-up construction forces users to apply operations per data value. Although it could be feasible to implement mechanisms to eliminate repetition through automation in constructive approaches (à la *programming by example* [21, 51, 119]), iVoLVER's current version does not allow this.

- ***Visual Representations as Controllers***
  iVoLVER is designed to blur the line between visual representations and their control and configuration. The dataset, controls, configurations, and visual elements all reside in its infinite canvas, and most widgets are both representations of data *and* allow its manipulation. In contrast, Tableau Desktop's pane-based interface separates manipulation and representation: the central visualization pane shows the results of operations spatially separated from the configuration areas of the interface. In addition, access to some functionality takes place through menus.

To investigate the consequences of these two differing approaches to visualization authoring, I conducted two studies with non-experts: an in-classroom pilot study and a more in-depth laboratory study.

4.4 In-classroom Pilot Study

The pilot study was run as part of an introductory course on information visualization in the School of Computer Science of the University of St Andrews. The course was taught by Dr Uta Hinrichs who, besides my supervisor, collaborated in the execution of this study. Because of this, I sometimes use plural pronouns such as "we" and "us" in the text below.

As part of this within-subjects study students were introduced to Tableau Desktop and iVoLVER in two subsequent tutorials that took place in weeks three and four of the 11-week module. The first tutorial took place immediately after a lecture on marks, visual variables, and mappings. Students completed three visualization exercises using each of the tools and provided feedback about their process, the character of visualizations they built, and their general experience with the tools. The following sections describe the methodology—which is shared with the main study—in detail below.

4.4.1 Participants

Out of 36 students enrolled in the class, 19 volunteered to participate in the study. Of these, eight did not complete both study sessions, leaving us with 11 participants (7 male, 4 female; age 21–26, median 22). The analysis of findings considered only data from these students. Participants had backgrounds in computer science,

Country	Production	Consumption	Population	Region
United States	250	200	3224	America
Saudi Arabia	350	1050	308	Asia
Russia	600	600	1440	Asia
China	800	850	13760	Asia
Canada	1100	300	360	America

Table 4.1: Oil Production & Consumption Dataset

information technology, language & culture, and statistics. All were novices in InfoVis with varying self-reported expertise: four had no prior knowledge about InfoVis; four were aware of InfoVis through the web and/or magazines without knowledge about its theory; and three had been actively following visualization topics but did not have practical application experience.

4.4.2 Study Procedure

The pilot study was divided into two sessions. In Session 1 all students of the course were split into two groups. One group was introduced to Tableau Desktop and the other to iVoLVER. In Session 2 (the following week), students switched tools. Each study session started with an introduction to Tableau Desktop or iVoLVER to the respective groups. Students then completed a set of tasks with the corresponding tool and dataset. The visualization exercise involved datasets with five data points and five attributes each: one on oil production and consumption levels of countries (Table 4.1) and another one about movies (Table 4.2). The tasks were to create three different visualizations addressing different aspects of the data. For the oil dataset we asked participants to show:

- ***V1:*** The oil production levels by country,
- ***V2:*** The countries and their respective regions, and
- ***V3:*** The oil production, consumption and population by country.

The tasks for the movies dataset were equivalent in the type and number of attributes; participants had to show:

- ***V1:*** The budget of each movie,
- ***V2:*** The genre of each movie, and
- ***V3:*** Each movie's budget, world-wide gross revenue, and duration.

Title	Budget	Gross	Duration	Genre
Jurassic World	186.73	266.27	124	Sci-Fi
Avatar	340.53	408.89	161	Fantasy
Titanic	340.52	628.17	195	Drama
Star Wars	545.32	138.55	135	Sci-Fi
The Avengers	637.89	494.93	143	Fantasy

Table 4.2: Movies Dataset

Students had 30 minutes to complete the tasks. To balance tasks, datasets and tools, participants in both groups saw the same datasets and tasks in the first session, and different (but equivalent in size and difficulty) datasets and tasks in the second.

After completing the tasks, participants filled out questionnaires about their satisfaction with the tool, the visualizations they created, and the rationale for their representation choices. They also rated the process of constructing visualizations with the tool they used in each session and listed its most useful and problematic features. Session 2 ended with a group discussion with all students, who compared their experiences with each tool and commented on their perceived benefits and drawbacks.

4.4.3 Data Collection & Analysis

We recorded 650 minutes of participants' visualization processes in the form of video screen captures. I analyzed the video through a qualitative video coding process as proposed by Heath et al. [79]. Specifically, I coded for visualization types, selected mappings, exploration of alternative designs, experimentation with features of the tool, and problems observed during the visualization authoring process.

Similarly, participants' statements from the questionnaires were categorized and analyzed using qualitative coding. Codes focused on design approaches, satisfaction level rationale, useful and problematic features, and benefits and limitations of the tools. The 17-minute group discussion at the end of Session 2 was audio-recorded and transcribed. The qualitative analysis of discussion statements focused on perceived differences, benefits and limitations of each tool.

4.4.4 Pilot Study Results

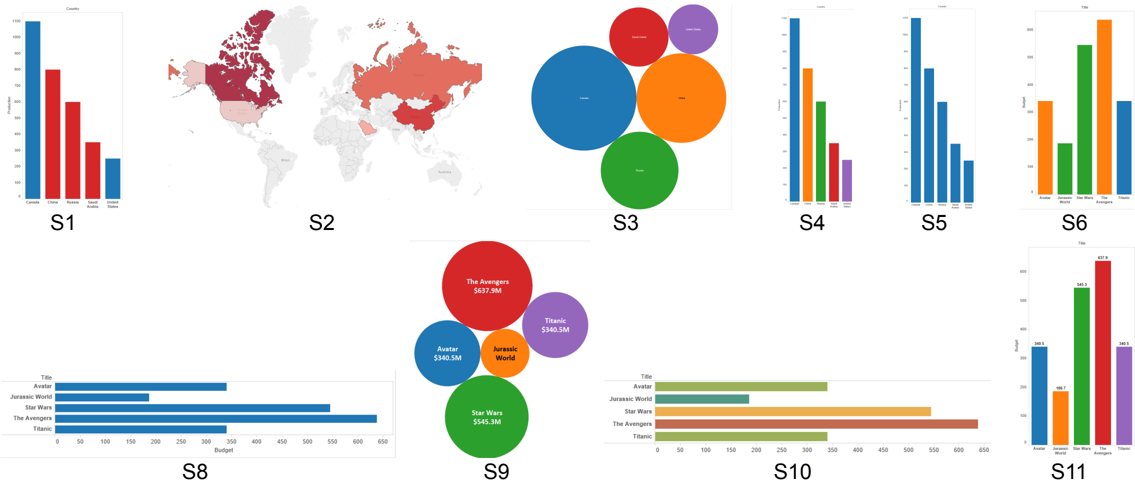
The analysis of the screen recordings gave us an initial understanding of the common sequences of actions participants carried out, the tool features they had problems with, and the extent to which they explored alternative designs.

Participants using Tableau Desktop explored more alternative designs than those using iVoLVER, but almost exclusively through the "Show-me" pane in something that appeared to be a trial-and-error approach. Also, the visualizations constructed in iVoLVER were more diverse than those created in Tableau and included non-standard visualizations (see Figures 4.2 and 4.3).

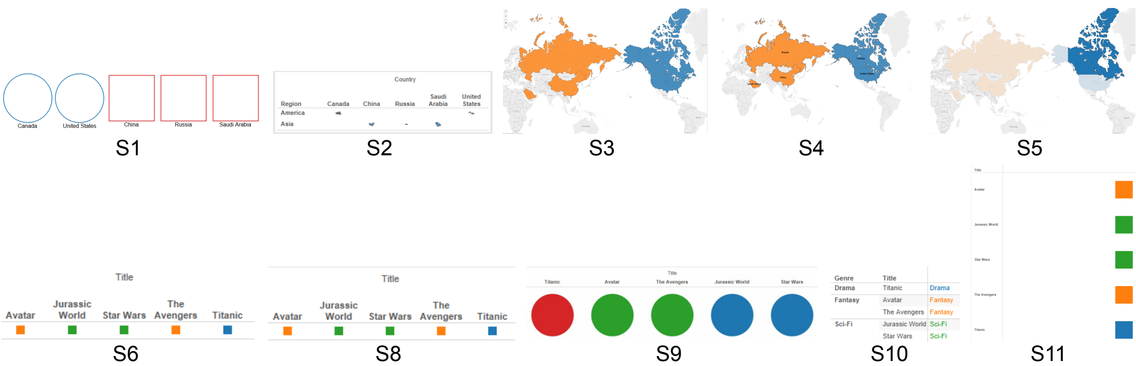
Some of the participants' motivations behind their design decisions could be inferred through their questionnaire responses: *"I just dragged and dropped the attributes and Tableau did the rest and I just kept moving them around until I found one I liked."* [S7], *"by exploring available chart types and then selecting them depending on the available properties/measures."* [S8, commenting about her visualization approach in Tableau]. While these responses and the video analysis suggested a fundamental difference in both tools regarding how students' approached their visualization design, the brief nature and vagueness of the students' written comments in the questionnaire did not provide enough context for an in-depth analysis of the nature of these differences. For example, when asked how they decided on the visual mappings, one student answered *"Messing around."* [S9], and another wrote *"Different shapes, I like circles."* [S2].

During the group discussion conducted at the end of Session 2 students highlighted the benefits and limitations of both tools. In particular, they identified trade-offs, for example, between the level of automation (as provided by Tableau) and flexibility (provided by iVoLVER) and the interplay between these features. However, only six students actively contributed to this discussion, so this data could not be considered as representative.

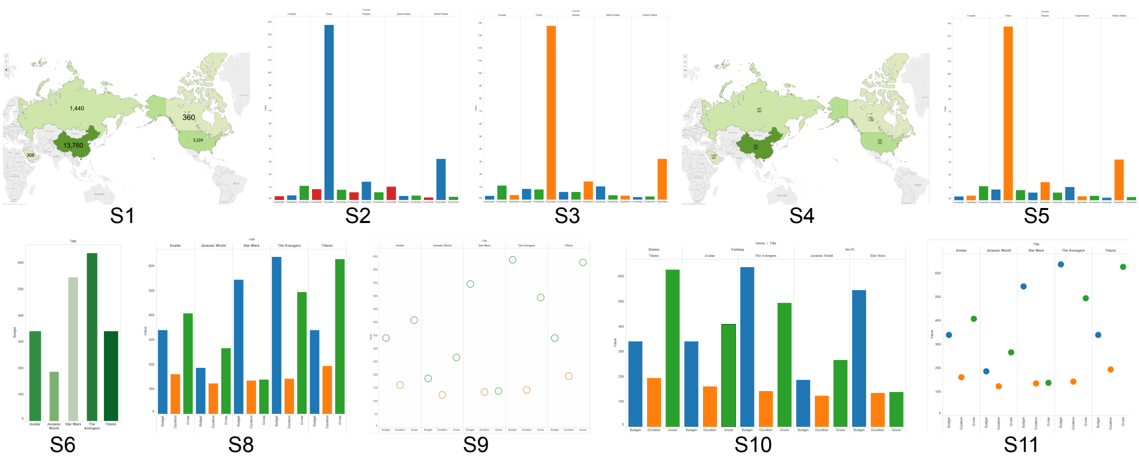
Perhaps more importantly than its results, the pilot motivated several key questions: What characterizes and drives the overall visualization design process using the two tools? Do top-down and bottom-up construction approaches influence the visualization design decisions and the exploration of alternative designs? If so, how? How do non-experts experience these two approaches in comparison? Following up on this pilot, we conducted a study to explore these questions more in-depth.



(a) V1

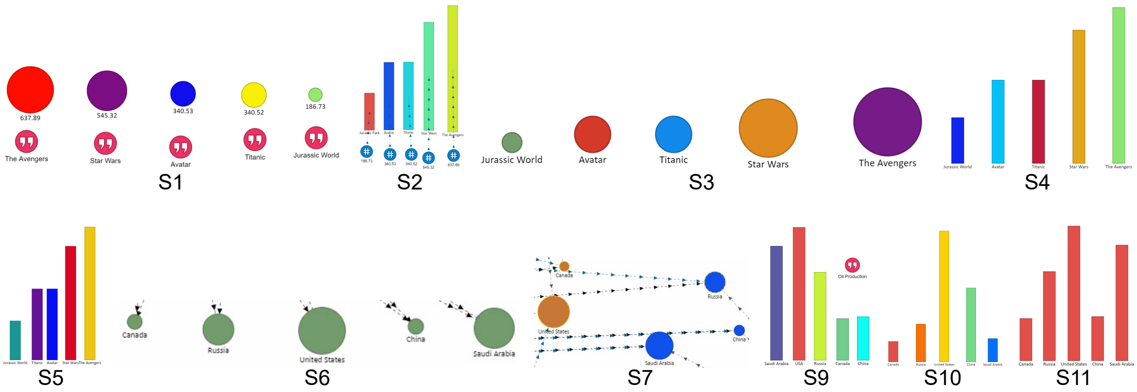


(b) V2

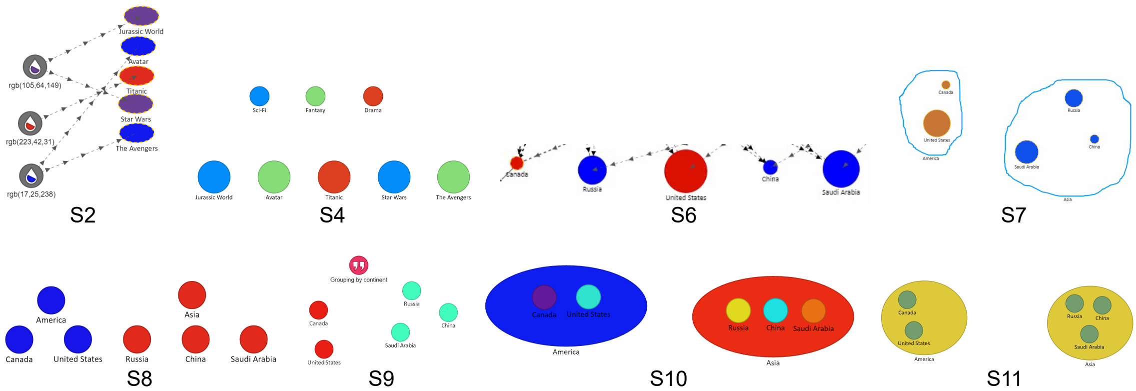


(c) V3

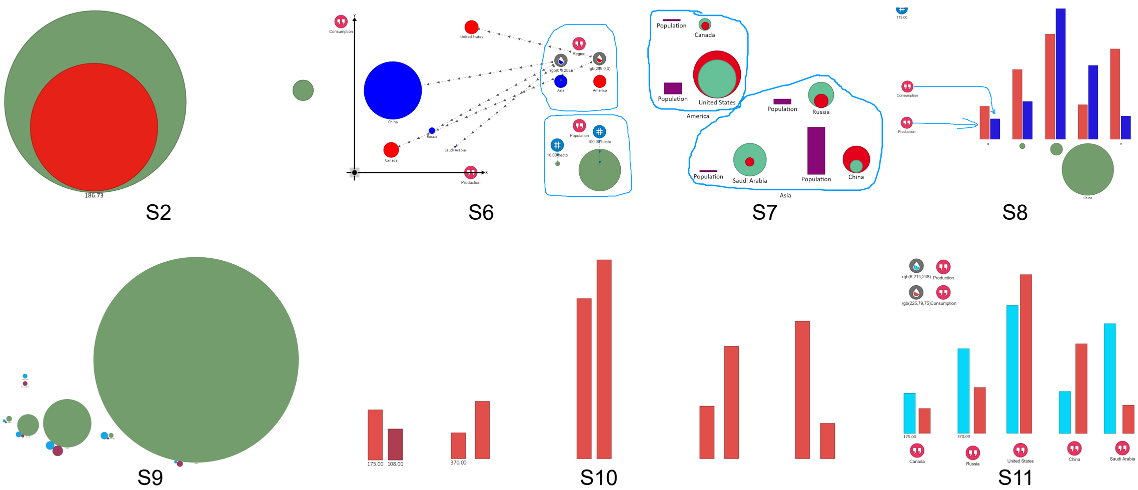
Figure 4.2: Visualizations built with Tableau by students of the pilot study labelled according to student number.



(a) V1



(b) V2



(c) V3

Figure 4.3: Visualizations built with iVoLVER by participants of the pilot study labelled according to student number.

4.5 In-Depth Comparison of iVoLVER & Tableau Desktop

This in-depth comparative study of iVoLVER and Tableau Desktop followed a similar procedure as the pilot, but it focused more on eliciting participants' individual perspectives on the tools which required one-on-one open-ended interviews. This study is therefore based on individual lab sessions, rather than an in-classroom setting.

4.5.1 Participants

Ten participants from the university of St Andrews were recruited via word-of-mouth (3 female, 7 male, 20–46 years old—median 28). None of them had participated in the pilot or ever taken an InfoVis course or any other training in visualization. Seven were undergrad or graduate students enrolled in computer science, mathematics, or physics; three were professionals with experience in teaching and/or researching in topics related to information technology, computer science, physics, and medical engineering. Two participants stated that they had no knowledge of visualization, six reported having come across visualizations on the web and/or in magazines, but never considered visualization concepts. One participant had been actively following and reading about visualization-related topics, but did not have any practical experience. Finally, one participant had been actively reading about the topic, and had experience in creating visualizations for at least one year.

All participants stated to be familiar with reading graphs and charts, with six stating to have created some, and one stating to be frequently involved in creating graphs and charts and to be quite comfortable with it. Some participants had experience with common visualization tools—such as MS Excel (10), D3.js (1), and Processing (1)—and programming languages, such as Matlab (4) and Python (5). None of the ten participants had used Tableau Desktop before the study, but four had heard of it. One participant had used iVoLVER once, six had heard of it but never used it, and three did not know it at all. No participant had ever received formal instruction on visualization concepts such as marks and visual properties. Even the participant who had used D3.js, stated that they had no conceptual knowledge on InfoVis. Participants received a £10 gift voucher as compensation for their time.

4.5.2 Study Procedure

Each study session consisted of the following activities:

- ***Introduction to Visualization Concepts:*** After filling out a questionnaire about their demographics and visualization expertise, each participant watched a 4-minute video explaining the fundamentals of the process of creating visual representations. The video explained the concepts of marks and visual variables and their relevance in the design and construction of visualizations. It also discussed how some visual variables are more (or less) effective for certain data attribute types.

  The video ended with a slide showing several visualizations encouraging participants to think about the composition of marks and their visual properties in these representations. Participants kept a printout summarizing the video to have at hand during the tasks (see Appendix B.1). This stage ensured that all participants had at least some minimum knowledge on basic visualization concepts, similar to participants in the pilot who had seen these concepts as part of their InfoVis course.

- ***Introduction to Tool:*** A 20-minute tool introduction session followed for the first tool (iVoLVER or Tableau Desktop) that included an overview of its interface and a demonstration of how the tool implements the concepts of marks and visual variables, as well as the interactions steps required to map data to visuals.

  This live demonstration showed participants how the tool was used to create a visualization of the small dataset shown in Table 4.3. Participants were provided with a printed "tool cheatsheet" to help them in the subsequent visualization tasks (available in Appendices B.3 and B.2).

Person	Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65

Table 4.3: People Dataset

- ***Visualization Task:*** Participants received a dataset and tasks to complete identical to those in the pilot study. Participants had 30 minutes to complete these tasks.
- ***Feedback on Tool:*** Participants filled out a questionnaire rating their experience with the visualization tool, their satisfaction with the resulting visualizations, and their understanding on the dataset after interacting with the tool. An open-ended interview followed in which they described their visualizations, their creation process, their mapping choices, and their general experience and satisfaction with the tool and its features.

Participants went again through the sequence of intro-to-tool, visualization task, and feedback session with the second tool and the second set of data and tasks. Tool order was counterbalanced across participants to reduce learning effects and assure that each tool was tested the same number of times on each dataset and task. The study session concluded with a final interview where participants compared how iVoLVER and Tableau Desktop supported the visualization process and listed their perspectives on benefits and limitations of each tool. They were also asked if they would use any of the tools in the future and if so, for what types of visualization tasks. Each study session took approximately two hours.

4.5.3 Data Collection & Analysis

Questionnaires were used to collect participants' demographic and background information as well as their ratings of the visualization process and results. The questionnaire answers were coded and categorized according to themes such as design ideas, rationale for the choice of visualization type, visual mapping strategies, identified benefits and limitations of each tool, and potential usage scenarios.

In each session I took written notes of participants' interaction processes with the tools and video captures of all on-screen activities. In total, 415 minutes of video data were analyzed using qualitative methods as proposed by Heath et al. [79]. This included a characterization of the individual visualization steps and a detailed analysis of particularly interesting episodes flagged in my observational notes (e.g., patterns in the use of specific functionality and problematic interaction paths). I also collected the visualization outcomes (30 in Tableau and 28 in iVoLVER—two participants did not complete all tasks due to time constraints).

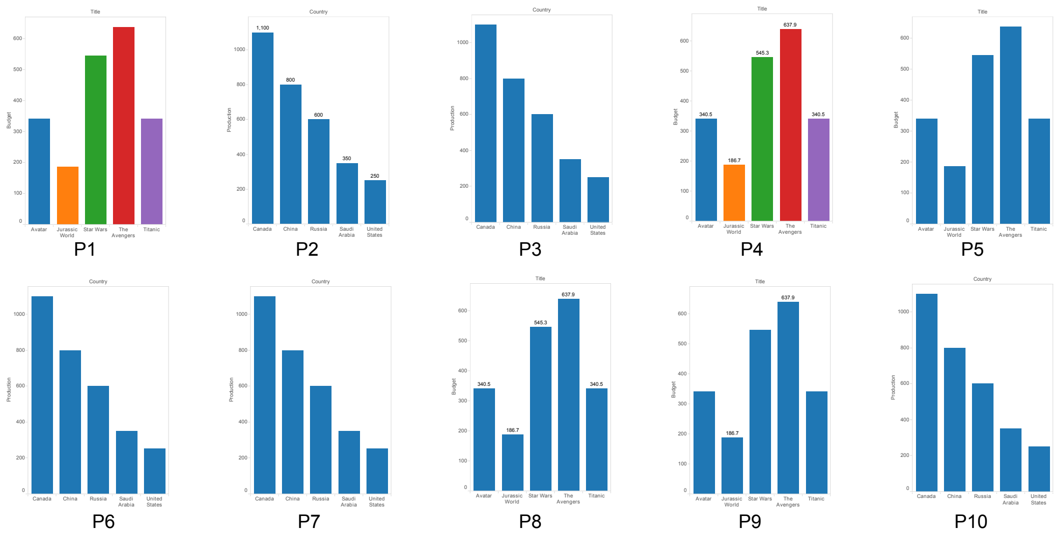
All interviews were audio recorded, fully transcribed, and qualitatively coded by myself following a thematic analysis approach [27]. For comparison purposes, Dr Uta Hinrichs also coded a small portion of the data in an independent process. Initial coding focused on themes from the interview questions such as the visual mapping process and the benefits and limitations of the tools. From there, higher-level themes emerged, for example, suitable usage scenarios for each tool, and perceived ownership of the process and results. The coding scheme evolved through regular meetings with Dr Hinrichs until the coding scheme stabilized.

4.6 Findings

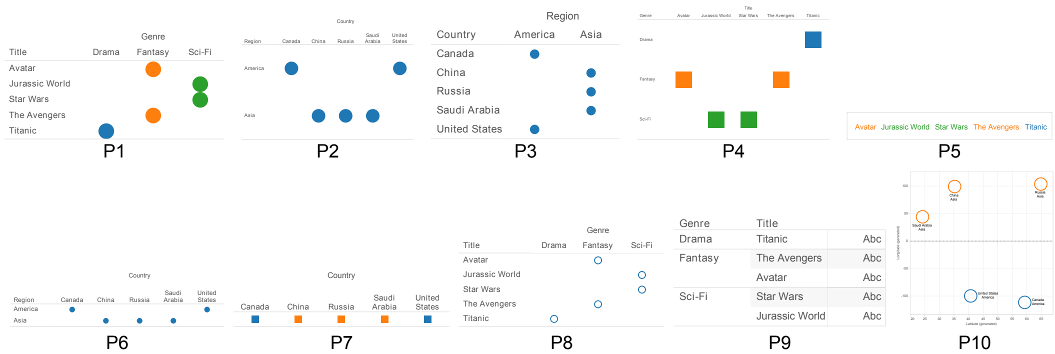
Figures 4.4 and 4.5 show the visualizations created with Tableau Desktop and iVoLVER, respectively. Visual inspection supports an overall difference in the range of visualization types created with each tool.

- ***iVoLVER:*** Participants created a large variety of visualization types for the three tasks. For V1, most participants (8/10) produced bar charts; two participants created a bubble chart. For V2 there was a strong preference for representing qualitative attributes with changes in shape and color (6/10); other visualization approaches included a Venn diagram (P1), a graph representation (P2), a table (P8), and a cluster-based visualization (P10). For V3, scatter plots constitute half of the visualizations, together with one multi-bar chart (i.e., one bar chart for each attribute—P8), one unfinished graph representation (P2), and three non-standard visualizations with attributes mapped to shape, color, and/or the width and height of rectangular marks (P3, P6, P9).
- ***Tableau Desktop:*** Visualizations created in Tableau Desktop form a less heterogeneous set. All participants created bar charts for V1. The majority of visualization types created for V2 are tabled-based representations with marks located in cells (7/10), with the exception of one visualization containing colored text marks (P5), one text table (P9), and a scatter plot where the latitude and longitude auto-generated by Tableau were used to position circular colored marks to represent different countries (P10). V3 visualizations were split between multi-bar charts (6/10) and scatter plots (4/10).

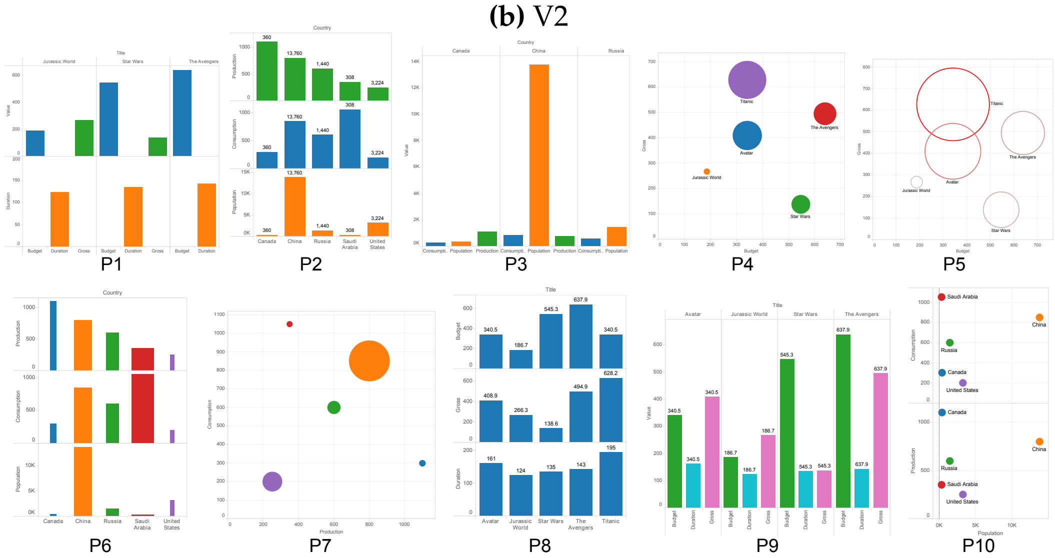
The popularity of bar-based constructions visible in the visualization results of both tools is consistent with Grammel et al.'s previous findings of novices constructing visualizations in Tableau with the assistance of an expert [71]. The entire set of



(a) V1

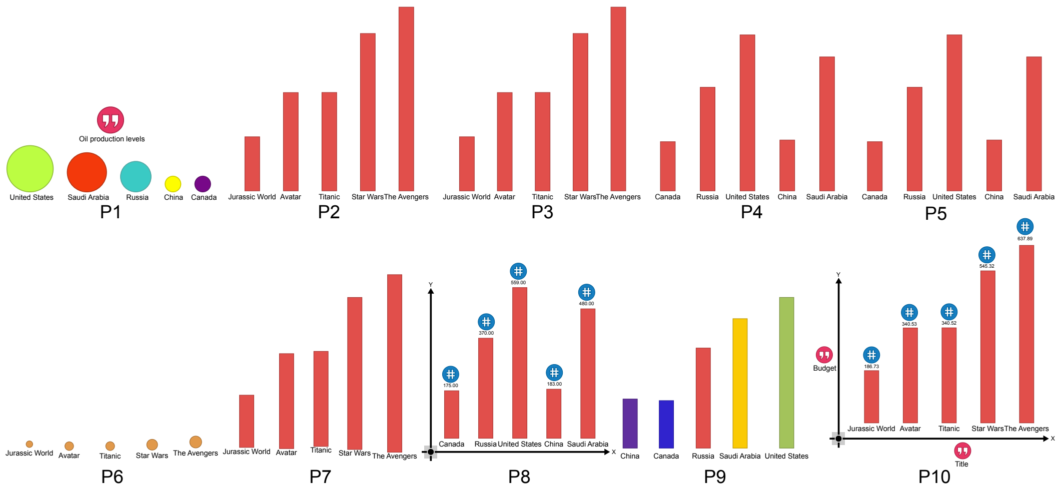


(b) V2

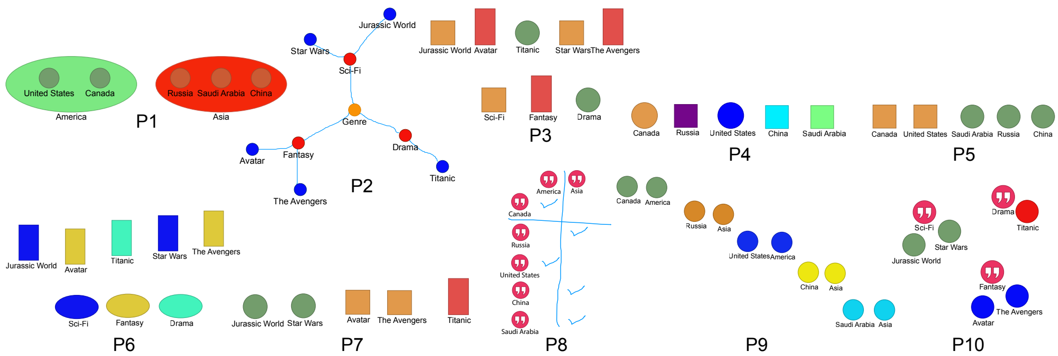


(c) V3

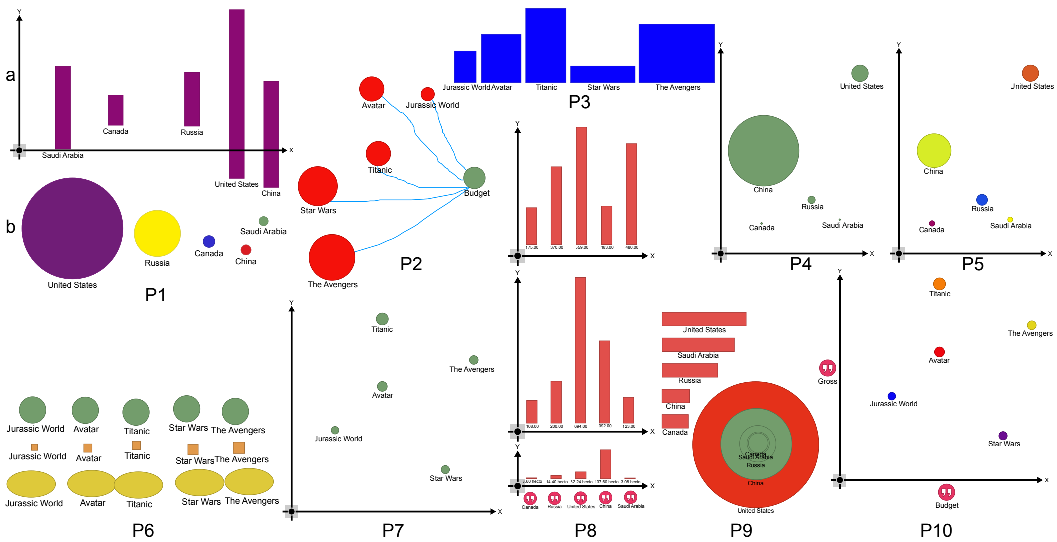
Figure 4.4: Visualizations built with Tableau Desktop by participants of the main study labelled according to participant number.



(a) V1



(b) V2



(c) V3

Figure 4.5: Visualizations built with iVoLVER by participants of the main study labelled according to participant number.

designs also resembles the *representation continuum* discussed by Walny et al. in the context of data sketching [217], but our participants did not create line graphs or pictorial representations.

4.6.1 The Visualization Construction Process

The analysis of the video and interview data revealed that participants approached their visualization construction process differently in iVoLVER and Tableau Desktop, as summarized in the sequence diagrams of Figure 4.6.

In ***iVoLVER***, participants followed a mostly linear process (see Figure 4.6a). They reported that they first identified the *data attributes* relevant for the given task and then tried to come up with an *idea* of the type of visualization they wanted to build (e.g., a bar chart) or the types of marks they would use to represent the data (e.g., circles): *"For the first design I just thought about all the general bar graphs that we see all the time in, you know, publications and stuff like that."* [P3].

Participants would then engage in a *representation process* trying to realize their initial idea. This process typically involved extracting relevant data points from the data table and mapping them directly onto the marks' properties. However, some participants also reported creating first some marks and assigning them colors or labels, before connecting these marks to data points: *"It's like you're pulling shapes into the thing [canvas] in the same way you might in a drawing package. And only then are you kind of introducing data aspects."* [P5].

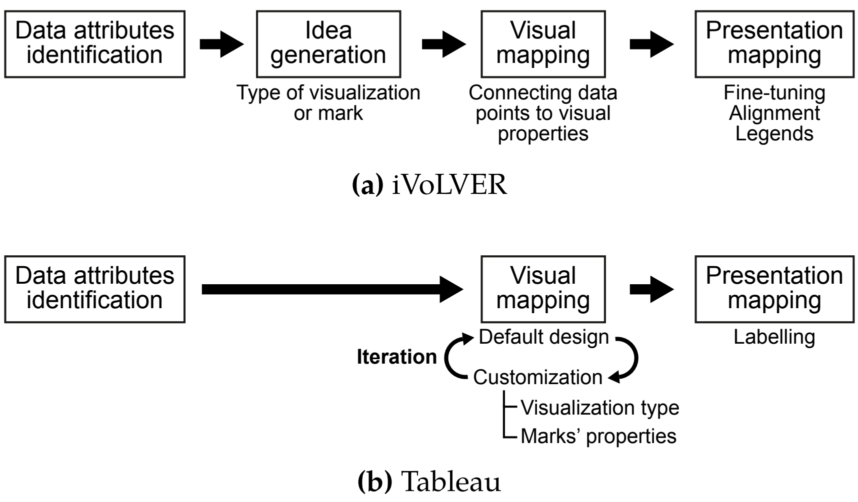


Figure 4.6: Stages of the observed visualization construction processes performed with iVoLVER and Tableau Desktop.

Applying textual labels to marks to refer their corresponding data—typically discussed as a step of presentation mapping [102]—was an integral part of the representation process, as it allowed participants to keep track of what mark represented which data point. Participants would also position, align, or group their marks intentionally. Final layout adjustments and the creation of legends (2/10) concluded participants' visualization construction process in iVoLVER.

What stands out in this process is its linearity. The exploration of alternative visualization designs or iterating on initial ideas was rare with only one participant attempting to construct different designs in iVoLVER (see P1 for V3 in Figure 4.5). Exploration was mostly done in the context of testing the suitability of a representation idea, for example, a particular mapping using a single data point and a single mark. Participants' statements indicate that it was the elaborate construction process that prevented them from further design iterations or explorations in iVoLVER:*"Many things need to be manually done. For example, even for these five marks I need to drag fifteen times. [...] But if, after I drag fifteen times, I realize 'no, a bar chart is not good' and I need to change, then, probably it would be quite difficult to change at that point."* [P8]. Comments of this type highlight the scalability problem of ConstructiveVis discussed in subsection 3.8.2 of Chapter 3. Strategies to overcome this problem are explored in Chapter 5 of this dissertation.

In ***Tableau Desktop*** (see Figure 4.6b), as with iVoLVER, participants first identified the *data attributes* relevant to their task. However, they then typically started the *representation process* right away, dragging the attributes of interest into the rows and columns shelves, even if they did not have an idea in mind about how to represent these attributes. Only two participants reported having an initial idea before they started to explore visualization types in Tableau Desktop. Participants typically followed Tableau's default suggestions for their initial visualization design (visible in the Tableau visualizations for V1 and V2 in Figure 4.4) and spent most of their representation process customizing this design (e.g., changing shapes or colors).

In contrast to iVoLVER, the representation process in Tableau Desktop was more iterative, with participants experimenting with the mapping of attributes to visual elements and rearranging the attributes in the "Columns and Rows" shelves to redefine existing mappings. Many participants also made use of the "Show me" pane to explore alternative visualization types, however, this type of exploration was much less frequently than in the classroom context of the pilot study. These notable differences in the sequence of actions confirm the results of previous studies that compared approaches of constructing bar charts using template-based digital tools and physical tiles [223].

4.6.2 The Tools' Influence on the Design Approach

When it comes to constructing visualizations, coming up with ideas on how to visually encode data is key. During the interviews, participants were prompted to comment on how they carried out their mappings and what drove this process. As shown in Figures 4.4 and 4.5, the two tools heavily influenced participants' approaches to visualization design.

With ***iVoLVER*** participants typically reported to have a design idea in mind and only then to start the practical process of mapping data to visuals. They often mentioned the need to plan and/or think before engaging into the representation process—this was experienced as challenging, but also encouraged creativity: *"In iVoLVER there is sort of an extra step where you actually have to come up with the plot itself and the strategy to build it. [...] When you say 'I wanna do bar charts' in iVoLVER, you have to think of stuff like 'OK, so... How am I going to order the bars?', 'How am I gonna connect?' It's up to you, all kind of different things: 'Am I gonna scale the data?'"* [P2]; *"[In iVoLVER] you have to think about the properties in one sense but also the marks. And it's slightly more creative because you are making that selection, before you're seeing a bar chart or a graph or something that's quite default and typical."* [P9].

In contrast, the exploration of designs in ***Tableau Desktop*** was tool-driven, as shown by the video analysis and interviews. The process was characterized by participants following Tableau's design lead in the absence of an initial idea or reflection on adequate data mappings: *"I mean, I don't really have to think about [the design] in any way. I just put the title in one column and the value in the rows and, you know, that automatically created the bar chart, which is perfectly adequate. [...] I didn't necessarily have a bar chart in mind. I just wanted to put it there and decide later on how it's best represented depending on what the program does. [...] And then see where to go from there."* [P1].

Six participants mentioned or suggested this notion of "no-need-to-think" when working in Tableau: *"I was just dragging values around like 'Oh, I'll put this in the columns', 'I'll put this in the rows', 'I'll change this mapping from this thing', but without really thinking about it."* [P5]; *"[Tableau] feels like a very quick and easy way of presenting some simple data if you don't have time to think about how you want to do it in a different* *way."* [P9]. One participant directly compared the design process of the two tools—their conclusions are representative: *"I really, really, liked the way that I can very quickly get something going [in Tableau]. I felt that [...] I didn't have to think as much about how to implement [the visualization] as I did in iVoLVER."* [P5]. These statements confirm what the video analyses of the construction processes in both the pilot and the main study hinted at: with Tableau Desktop participants experimented rapidly with a range of different design approaches, but typically without much critical reflection.

The impact of this tool-driven approach seen with Tableau Desktop should not be underestimated; the collected evidence suggests that it can fundamentally influence the way in which people think about visualization design—even if they do not use the tool any longer. Three out of five participants who worked first with Tableau stated that their decisions on visualizations types in iVoLVER were based on what they did with Tableau: *"Because I did the previous [task] using Tableau. The datasets are quite similar, so I already had in my mind it should be something like a bar chart, because I did the previous [task]."* [P8]. In fact, when working with iVoLVER, this participant replicated all the visualizations he previously built with Tableau Desktop. The study did not provide any evidence that working with iVoLVER influenced design choices in Tableau.

4.6.2.1 Design Exploration

The fundamentally different approaches to visualization design represented by the tools also have an influence on how and how much participants explored alternative design ideas. The speedy, tool-driven process in ***Tableau Desktop*** promoted the exploration of multiple mappings as indicated by the following statements: *"Since there are so many [design] options, you want to try putting data in different slots and try different kinds of graphs in order to see what looks better. In a way that's good because you get to explore more stuff."* [P3]; *"I can change from bars to other things quite easily. [...] so, I can quickly experiment with different charts and see the final one. But for iVoLVER I need to decide first and then I choose the bar chart."* [P8]. As mentioned earlier, with iVoLVER participants typically did not iterate on their visualization designs once they had built them, neither did they explore different visualization types (see Figure 4.6).

However, the type of exploration promoted by Tableau seems less in-depth compared to iVoLVER, with participants not critically reflecting on the meaning and potential impact of their explorations: *"There wasn't any need to do anything else because what [Tableau] did automatically was what I should have done, I suppose."* [P1]; *"Because it takes care of so much for you, you sort of have to build up a good set of experience to use it. You know, to just be able to expect what it will do."* [P2]. Furthermore, the tool-driven exploration hampered participants' experimentation with their own ideas: *"I think because it's quite automated, sometimes it's perhaps a bit more difficult to customize things in the way that you want to."* [P1]; *"It felt like I wasn't really composing the thing. I was kind of putting stuff into pre-canned visualizations."* [P5].

While design explorations can be laborious in iVoLVER, they allow for more free-form, creative experimentation and critical reflection. iVoLVER seems to promote a more "thoughtful" approach to visualization design exploration: *"I think the process [in iVoLVER] is very much an atomic thing. So, you do know 'OK, this one is bigger than this one', because you're dragging it and you're doing it. And so you're thinking about the values of each thing you are putting on consciously rather than just assuming."* [P10].

4.6.2.2 Deciding on Visual Mappings

The visualization design process not only includes coming up with an idea of how to represent the data as a whole (e.g., the type of visualization) but also how to encode particular attributes using visual variables such as size, color, or shape. According to participants' statements these low-level visual mappings were typically a consequence of the visualization type chosen: *"I chose bar charts and, of course, the height is the suitable [visual variable]."* [P8, about his process with iVoLVER]. However, most participants, regardless of the tool, were able to justify their decisions on visual mappings by reflecting on the suitability of certain visual variables to encode particular attribute types. For example, working with Tableau, participants stated things like: *"The budget is quantitative, so I used length."* [P4]. Statements after using iVoLVER go in a similar direction: *"I decided on color because it seemed like a good way to represent categorical data."* [P5]. These statements, which link to what participants learned through our introductory video, indicate that in both tools they were able to apply visual mappings they came up with for their visualizations.

4.6.3 On the Tools' Underlying Authoring Approaches

iVoLVER and Tableau Desktop represent opposite extremes in the bottom-up/top-down visualization construction continuum described in section 4.2, and the consequences are reflected in the interviews.

Participants discussed iVoLVER's bottom-up strategy as laborious because it requires manipulating every single data value: *"For every circle I had to do the same, the same, the same thing. So, if there was a way [in which] I just do it for one circle, and it automatically does it for all the circles around—it would be perfect."* [P7]. However, they also highlighted flexibility and freedom as benefits of the bottom-up approach: *"Because of the way iVoLVER presents all the values you can change all of the direct mappings. I can kind of be a bit more creative with them. [...] Perhaps a good example would be that in the case of the third activity it was very, very easy to see that I could map one value to the height and one value to the width."* [P3]; *"[iVoLVER] was actually much more free and helped me come up with new ideas. In Tableau I had no idea what to do with that type of [categorical] data and there wasn't a huge amount of freedom to play."* [P10].

The above statements highlight an important trade-off: Tableau's top-down approach results in very few required interactions allowing for the speedy creation and exploration of different visualization types. However, this comes at the cost of transparency of the visualization process. When working with Tableau, it was sometimes difficult for participants to follow how data was mapped to the visual properties of marks, in contrast to iVoLVER where this connection is clearly visible. Along these lines, one participant stated: *"iVoLVER is better for design because you actually get to see what works, what doesn't work and why. Whereas in Tableau I don't think you get to see what works and doesn't work and why [...] because it's generated so quickly that you don't get the sense of something building up. Whereas in iVoLVER you've built something up from a quite low level. Whereas in Tableau you just come in and something is created. You don't see how it is created or why."* [P10].

In general, most visualization features in Tableau operate on an attribute-level. User interactions (e.g., dragging data attributes to shelves) can introduce fundamental changes in the visualization that can be difficult to follow. Commenting on their experience with the "Show me" pane, one participant stated: *"At one point I clicked on it, and it changed everything. I was like 'This is a disaster!' [...] I was just not quite sure what it was going to do once I put genre in columns. It was just very abstract."* [P4]. This, along with previous statements, makes clear that predicting, keeping track of, and interpreting exactly how Tableau Desktop maps data to visual properties was often difficult for participants due to the speed and the attribute-level scale in which things change based on user's interactions. That being said, the links that show the mappings between data and visuals in iVoLVER, while adding transparency, also frequently cluttered the interface and, therefore, hampered the visualization process. Four participants mentioned this explicitly: *"You get to a certain level of complexity [in which] it becomes very hard to manage all connections."* [P5]; *"You have the connections, which, again, are quite confusing sometimes. Again, you have to keep good track of what you have done so far."* [P1].

4.6.4 Control and Authorship

The type of words participants used when describing their visualization process in Tableau vs. iVoLVER further indicates that the visualization process in Tableau is largely tool-driven. When explaining their iVoLVER constructions, all participants assumed authorship of the process and outcomes. This was visible in the use of first person singular personal pronouns (e.g., *"I did"*, *"I mapped"*), as the following statement illustrates: *"And the third [visualization] I'm satisfied with because I showed different data in, like, a readable way."* [P4].

Across all interviews, in just a single instance iVoLVER was credited with a design decision: when a participant released a connection on top of a circular mark and iVoLVER associated this to the mark's radius property (its default behavior). In contrast, in their descriptions of the process with Tableau most participants (8/10) at some point accredited Tableau the responsibility for certain design decisions: *"Tableau did a very good job of ordering them. For example, now we know what Canada produces."* [P6]. It was also common for participants to describe their visualization results as a consequence of a mix of their own and Tableau's actions which they clearly separated, as illustrated in this exemplary statement: *"I dragged the title and I dragged the gross here. Then it [Tableau] automatically generated this [visualization], which I think is good enough. I can understand this already and, again, I wanted to add the labels to make it easier to read. [...] so, I tried to drag the label."* [P8]).

Authorship was also explicitly commented on (without being unprompted) by six participants when summarizing their experience with the two tools: *"[In] Tableau you just throw it [the data] at the system; throw everything and then it generates [the visualization] for you"* [P8]. And commenting on iVoLVER: *"I'd say that the process [in iVoLVER] is definitely more involved in that you have to actually interact with that data* *instead of just throwing the idea of a column into a thing and having it built for you. [...] Because designing is something that you need to be involved in, and you need to understand what's happening with the data for you to show what's happening with the data."* [P4].

4.6.5 Overall Preferences

At the end of each study session, participants were asked what tool they preferred overall. Most of them favored Tableau Desktop (7/10) because of the speed of construction. However, two of these seven also mentioned a trade-off between speed and: (1) opportunities to explore (*"I did everything quicker in Tableau, even though there was less space for exploration."* [P3]; and (2) experiencing the process (*"Because I am writing papers, I just want to see the result; I don't need the process. So, Tableau would be better. But if sometimes I need the process I might choose iVoLVER."* [P8]).

One participant preferred iVoLVER because of its flexibility, but considered Tableau her default option because of its speed: *"I'm a very visual user. So, although I got a bit stuck with iVoLVER, I would probably persevere with it just because it let me drag and play around and I could quickly see changing shapes and objects and things like that. But I could see me defaulting to Tableau if I have to quickly create a graph."* [P9].

As indicated by these statements, all participants found the tools' suitability to be context dependent. They described Tableau as convenient to deal with large datasets: *"I'd probably prefer Tableau because I generally use large datasets, and in a large dataset iVoLVER is just not gonna handle it. And so, Tableau would be the only viable option."* [P10]. iVoLVER was considered useful for prototyping, for exploring unfamiliar datasets, and for more creative approaches to visualization: *"iVoLVER could be a really powerful tool to prototype visualizations because of its generality and because it's really convenient and really transparent."* [P2]; *"I would say [iVoLVER] is much more useful probably with more unique types of visualizations. Something that isn't just a standard chart; something where you want to have that fine grained control; and something where you aren't super familiar with the dataset and you want to figure it out as you go."* [P4].

Two participants mentioned a hybrid approach using both tools. They identified iVoLVER as a good starting point to experiment with potential designs and Tableau Desktop as preferable for implementing the visualization once the design has been decided: *"If I don't know what I want yet, I would prefer to try iVoLVER first to see in how many ways I can design the visualization. And then, if I know what I actually want* *already, I would just put it on Tableau because it's faster."* [P7]; *"I would use iVoLVER for quick drafts of data visualizations. And then if I had to do maybe a more professional one, then maybe move to Tableau."* [P3].

4.6.6 Teaching and Learning InfoVis

Because of its predefined templates, one participant mentioned Tableau as a good tool to teach people the design space of standard visualizations: *"If you're sort of learning a taxonomy of different types of visualizations, Tableau is probably better because they already exist, and you can experiment very quickly by just moving a few things around."* [P5].

Conversely, iVoLVER was mentioned as useful to train beginners because of its transparency regarding the visualization process which brings across basic visualization concepts: *"If I was trying to show people the ideas and the intuition and the motivation behind visualizing things, definitely iVoLVER. Once again, the triviality and the transparency [of the process] are the two winning factors."* [P2]; *"Because it's more intuitive. At least I believe that in order to learn something, you have to get your hands dirty; you actually have to go and see. The level of automation of Tableau is very high. It's just like drag, and then you have no idea what would happen. For iVoLVER you can influence directly what will happen. [...] I believe that for a beginner iVoLVER is better than Tableau."* [P7]; *"Tableau does it for you, which isn't helpful at any kind of education system. Because you don't have to figure out what specifically is happening and how position works and how area works. It's just sort of you drag it into size and—whoop!—it does it. Whereas in iVoLVER you need to actually pay attention."* [P4].

4.7 Discussion

The study findings show important differences between iVoLVER and Tableau Desktop in outcomes, process, design approach, authorship, and preference. Participants recognized Tableau Desktop as an efficient tool, but often used it in a haphazard way—"no thinking required". Although the visualization process in Tableau Desktop included more alternatives, the visualization outcomes were also more homogeneous. The analysis also shows that participants using iVoLVER initiated the design process with their own ideas whereas Tableau Desktop users were strongly led by the tool's design suggestions. Although most participants preferred Tableau for its speed (7/10), the consensus was that iVoLVER facilitated a better understanding of the visualization process and promoted a stronger feeling of control and authorship.

4.7.1 Efficiency vs. Understanding, Freedom and Creativity

The comparison of the two tools highlights the consequences of bottom-up and top-down visualization authoring approaches on process and outcomes, which is relevant for designers of visualization tools for non-experts. The trade-offs are remarkably tricky; Tableau's top-down approach allows people to quickly explore many alternative designs compared to the more laborious bottom-up constructive process of iVoLVER. This reflects the tension between magic and literalism identified long ago by Smith [190], but it goes further: the efficiency of the top-down template-based or shelf configuration approaches might come at the cost of thoughtfulness and affect the transparency of the process. When participants are forced to construct a visualization from the ground up (i.e., from the level of data points), they seem to be involved in a deeper process that results in a better understanding of how the visualization evolves and in a feeling of control, ultimately leading to more creative and varied designs and a stronger experience of authorship.

In many situations tools need to be, first and foremost, quick and efficient; however, it is important to recognize that visualization design is also a creative task. There is value in enhancing the breadth of people's explorations for visualizations that support their own purposes. Similarly, promoting "thoughtful exploration" and a thorough understanding of the consequences of representation design decisions is more in line with the goals of visual literacy [26, 87, 129], and is an important part in educational contexts. This is in stark contrast to "black boxing" functionality that might leave important design decisions to the tool.

4.7.2 Scalability of the Constructive Process

As discussed in section 3.8.2 of Chapter 3, several of iVoLVER's design principles (atomicity, visual explicitness, and no menus or hidden operations) have a clear impact in the scalability of the visualization design process (i.e., visualization of larger datasets is tedious and requires a lot of repetitive steps and effort from the user). iVoLVER's scalability problem was also clear in the execution of the studies reported in this chapter, as evidenced by some of the participants' comments. Not being able to scale well to larger datasets prevented users from exploring alternative designs in iVoLVER (something that did not happen when they worked with Tableau Desktop). This could limit the usefulness of the tool in real-world analytical scenarios. Solutions to the “scalability problem” of ConstructiveVis are investigated in Chapter 5 of this dissertation.

New operations, such as an abstract function composition functionality, and interaction extensions such as macros could probably help. However, these solutions might also affect the tool’s expressiveness and would probably translate into a more complex environment and required interaction (against the simplicity principle of ConstructiveVis).

4.7.3 The Space in Between

Tableau Desktop and iVoLVER represent two extremes of a spectrum of visualization authoring approaches. Tools that allow “switching” between these two philosophies, or hybrid tools that bridge the idea of template-based, shelf configuration and constructive strategies may maximize the benefits and overcome the disadvantages of each approach. However, finding a sweet spot that balances the trade-offs found in the studies reported in this chapter requires more research. Although it would be possible to speed up iVoLVER with more abstract dataset operations, this very act of repetitive hands-on data mapping might be what provides the observed sense of thoughtfulness, creativity and authorship. Additionally, increasing automation might reduce the transparency of the process and harm its understanding [91]. The design space in-between seems a promising avenue for subsequent investigations, and is explored in Chapter 5.

4.8 Limitations and Open Questions

The design choices of the pilot and the main study obviously impact the generalizability and validity of the presented results. First, I have studied only two particular tools (with their own idiosyncrasies and workflows) that I believe represent the two ends of the top-down/bottom-up spectrum. Second, carrying out a controlled experiment with particular datasets and tasks may have affected participants’ process and visualization approach. While this effect has to be considered when extrapolating the findings, I believe that most of the observations here reported can be linked to the tools’ underlying approach rather than to the specifics of the study. For example, participants highlighted the importance of the visibility and flexibility of the visualization design process regardless of the particular implementation of each tool. Participants commented—without prompting—on how the tools would scale (or not) with bigger datasets. Nevertheless, additional studies are required to investigate these research questions considering additional visualization tools, datasets and analysis tasks.

It should also be highlighted that all the participants of the study had a science background and therefore arguably an affinity to quantitative data analysis, which may have influenced the study outcomes. Additional studies should investigate if and how the findings expand to different populations (e.g., with a background in the arts or humanities) and age groups (e.g., teenagers and seniors). I believe that these new questions need to be addressed through further studies which, in combination with advances in the design and implementation of new visualization tools, will offer a more definitive emerging picture.

4.9 Conclusion

This chapter presented a two-part qualitative study that compared how non-experts design and implement visual representations of structured datasets with iVoLVER and Tableau Desktop. The chapter discussed these two tools as representatives of the ends of a continuum of strategies to visualization authoring spanned between bottom-up and top-down approaches. The presented study identified differences in how the tools' underlying approaches shape non-experts' visualization process, decisions and justifications regarding the visual representation of data, and their feeling of control and authorship toward the visualization outcomes.

The insights of this chapter uncovered the advantages and limitations of iVoLVER's constructive approach to visualization authoring. As a digital constructive visualization tool, iVoLVER promotes a more hands-on, manual approach to visualization that forces designers to do more themselves, which can result in a better understanding of visualization principles and values. However, the tool does not scale well to larger datasets because it requires a large number of repetitive interaction steps, which can affect how and how much users explore alternative designs. The results discussed in this chapter can inform the design of future visualization tools implementing constructive principles, that are fast and easy to use but also empower people to create effective visualizations and promote critical thinking on their design process.

Chapter Five

Exploring the Space in Between

The purpose of visualization is insight, not pictures.

— Ben Shneiderman [35, p. 6]

The comparative study with iVoLVER and Tableau described in the previous chapter identified several trade-offs that arise as consequence of a tool's underlying approach to visualization authoring. While constructive "bottom-up" tools (such as iVoLVER) promote a hands-on, user-driven design process that enables a deep understanding and control of the visual mappings, more automated "top-down" tools (like Tableau Desktop) are way more efficient and allow non-experts to rapidly explore complex alternative designs, but often at the cost of transparency.

This chapter investigates how to reconcile automation and computer-assisted design with the benefits of ConstructiveVis (Q2.C). In doing so, the chapter explores how to overcome ConstructiveVis' scalability problem (described in section 3.8.2) to support a visualization design process that is not affected by the size of the involved dataset.

To gain an understanding of this design space, I conducted an iterative design process and organized a series of design workshops with people of different backgrounds and expertise levels. The combined set of designs contains a variety of solutions that range from example-based approaches (that directly expand ConstructiveVis) to solutions in which the visualization tool infers the designers' goals and applies predefined design principles. After a detailed analysis of these solutions and some existing visualization tools this chapter provides an initial step towards designing scalable digital constructive visualization tools.

This chapter first discusses relevant work on automation & agency, and abstraction & encapsulation in the context of visualization tools. It then presents the outcomes of an iterative design effort, which is followed by a detailed description of the results from the design workshops. Finally, it presents a design space described in terms of agency and atomicity as dimensions that can drive the design of new visualization tools.

5.1 Background and Related Work

This chapter draws on the concepts of automation & agency, and abstraction & encapsulation in the context of visualization tools as discussed below.

5.1.1 Automation & Agency

Automation is an important concept in visualization tools in order to facilitate visual mapping of large datasets and to enable the visualization designer to rapidly explore different visual representations. For example, many tools feature recommendation modules that suggest designs as users manipulate data elements (e.g., Tableau's "Show Me" [126] and similar strategies in other visualization tools [25, 69, 156, 176, 221]). In general, automation is meant to ease tasks that would be otherwise unnecessarily difficult, repetitive, or tedious. However, research from psychology and cognitive sciences has shown that too much automation may lead to complacency and bias [154, 153]. People tend to trust computers quickly and, thus, are less likely to reflect on what automated routines actually do on their behalf [52]. Furthermore, people often give more weight to the computer's decisions even in the presence of contradicting evidence [37].

In digital tools, excessive automation can also affect how people perceive agency, a concept that *"refers to a person's ability to control their actions and, through them, events in the external world."* [74, p. 242]. The experience of agency is an important aspect of HCI research because users *"strongly desire the sense that they are in charge of the system and that the system responds to their actions."* [181, p. 75]. Coyle et al. [49] found that too much computer assistance can harm people's sense of agency in a point-and-click task. As shown in the previous chapter, this can also happen in visualization tools: people experience different levels of control and authorship of the visualization design process when working with tools that offer opposite levels of automation (e.g., Tableau Desktop vs. iVoLVER).

Other research efforts within the visualization community have also paid attention to agency. Tori and Möller's taxonomy, for example, emphasizes the human aspect of visualization techniques by considering *"how much the designer chooses display attributes"* [206, p. 154]. Similarly, Koytek et al. incorporated the concept of agency into brushing and linking interaction [113]. Yu and Blackwell investigated the role of timing on user's agency in mixed-initiative interaction [225]. Other works have also looked at the combination of algorithmic and computational power with manual operations (e.g., [7, 141, 155]). This chapter specifically considers how to support both agency and automation in visualization tools.

5.1.2 Abstraction & Encapsulation

To deal with datasets of different types and scales, visualization tools often create abstractions of the data and visual mapping processes. While constructive approaches have a low level of abstraction (as values are directly mapped to visual properties), other tools implement different types of abstraction on a data, visual, or interaction level. Abstraction is a cognitive process that plays an important role in human language and thought [86, p. 184]. It is a complex term discussed across many areas of study such as philosophy [162], psychology [59, 189] and mathematics [78], and it is used in engineering to cope with complexity [110], although it necessarily implies hiding of detail (i.e., encapsulation).

The remainder of this chapter will refer to one specific type of abstraction—generalization abstraction—defined narrowly as the ability to refer to a set of items with a symbol. For example, when dealing with data, the name of a column becomes an abstraction of all the values contained in that column (i.e., all the values associated to an attribute). Abstraction can also be recursive: one can abstract a group of abstractions (e.g., one could refer to the group of `Name`, `Surname`, and `Age` columns as the `Person` data table, which raises the level of abstraction). The definition of abstraction used in this chapter is related to encapsulation in software engineering and object-oriented programming [192], and it has been discussed from the point of view of visualization of algorithms [140].

5.2 Goals and Methodology

The aim of the research described in this chapter is to explore how to bring together the benefits of bottom-up and top-down approaches to visualization tools. As suggested by the findings presented in Chapter 4, this is a difficult challenge as the benefits of bottom-up visualization tools—a deep involvement and understanding of the visualization process—might be incompatible with the benefits of top-down visualization tools: the quick and easy development of visualizations even of large datasets, made possible through automation.

To investigate possible solutions to this challenge in-depth and from multiple perspectives, I followed a mixed design methodology driven by: (1) my own and collaborators' experience in designing visualization tools and UIs, and (2) by the ideas from people with varying degrees of expertise in data visualization and visualization tools, who took part in four design workshops. In the search of design approaches to visualization tools that would be suitable not only to domain experts but also to visualization novices, the involvement of a larger group of people with different backgrounds was important as it led to a rich set of solutions. These explorations were done with iVoLVER and focused on revising this tool's design to make it scalable (i.e., suitable for visualizing larger datasets) while maintaining transparency and user-involvement in the visualization process. The following sections first present the in-house designs. This is followed by a discussion of the design workshops and the participants' solutions.

5.3 In-house Design Solutions

Before the execution of the design workshops, I sketched my own solutions to the problem of visualizing larger datasets in iVoLVER. I then engaged in an iterative process with Dr Miguel A. Nacenta in which we revised these designs. We held four meetings of about 1.5 hours each in which we incrementally refined my initial ideas using paper sketches.

5.3.1 Macro Recorder

In this solution, shown in Figure 5.1, users specify visual mappings by connecting individual values of the dataset (i.e., cells of the data table) to a mark's visual properties while an additional interface element, the *Macro Recorder*, registers the history of manual actions throughout the visualization process. The recording is activated upon request, and when finished it allows to select a subset of the performed actions and a set of data records (i.e., rows of the dataset) on which the selected actions will be applied. This generalization step results in a set of new marks that represent the selected data according to the mappings specified by the actions on the initial mark.

This design is based on the concept of demonstrational interfaces that *"let the user perform actions on concrete example objects [...] while constructing an abstract program"* [145, p. 61]. It preserves iVoLVER's bottom-up approach as the visualization design process takes place by specifying mappings at the level of

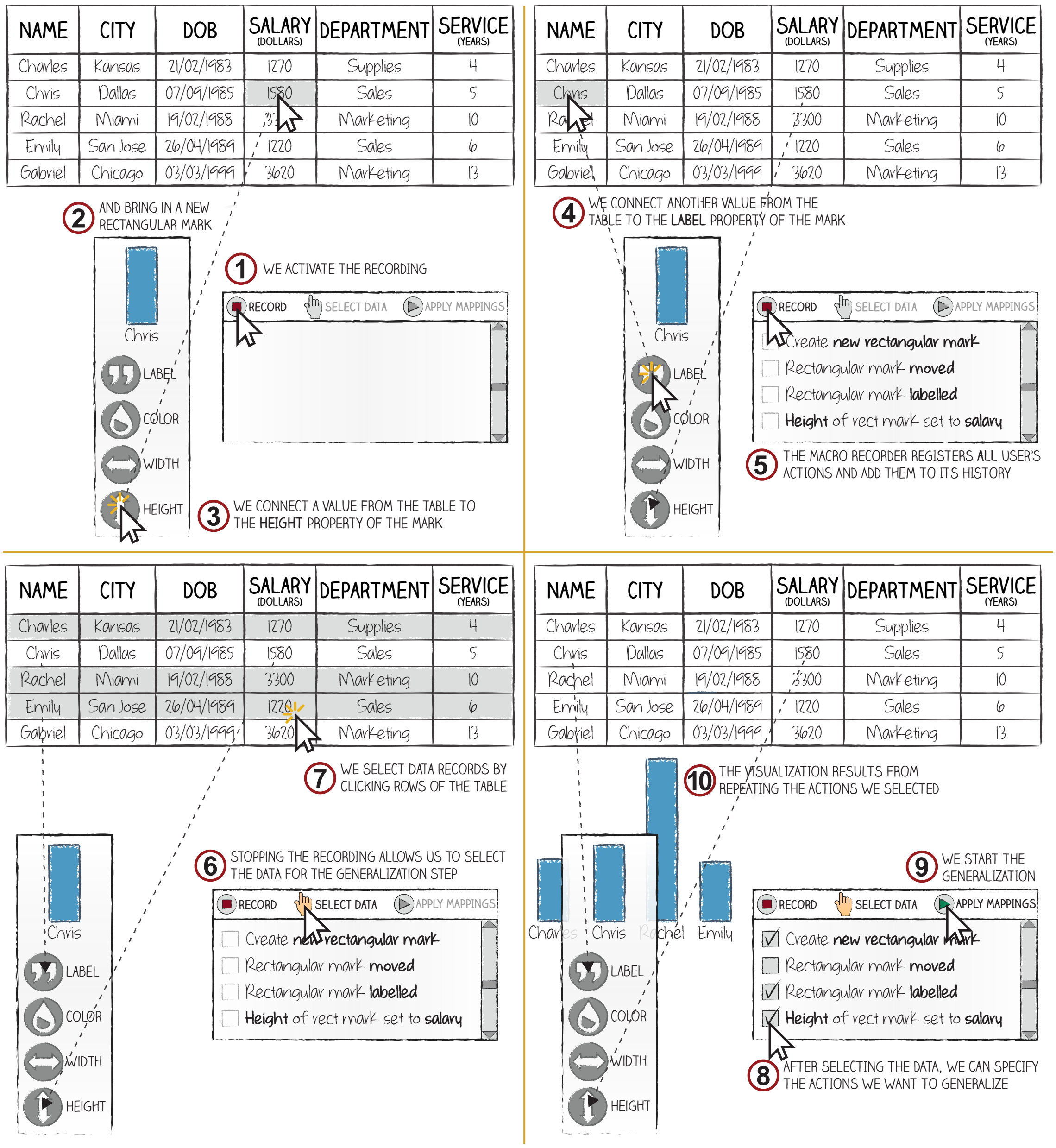


Figure 5.1: In-house Macro Recorder

individual data values and marks. The recording feature allows for generalization and, in this way, resolves the problem of having to repeat individual interaction steps, one of iVoLVER's main drawbacks.

5.3.2 Smart Assistant

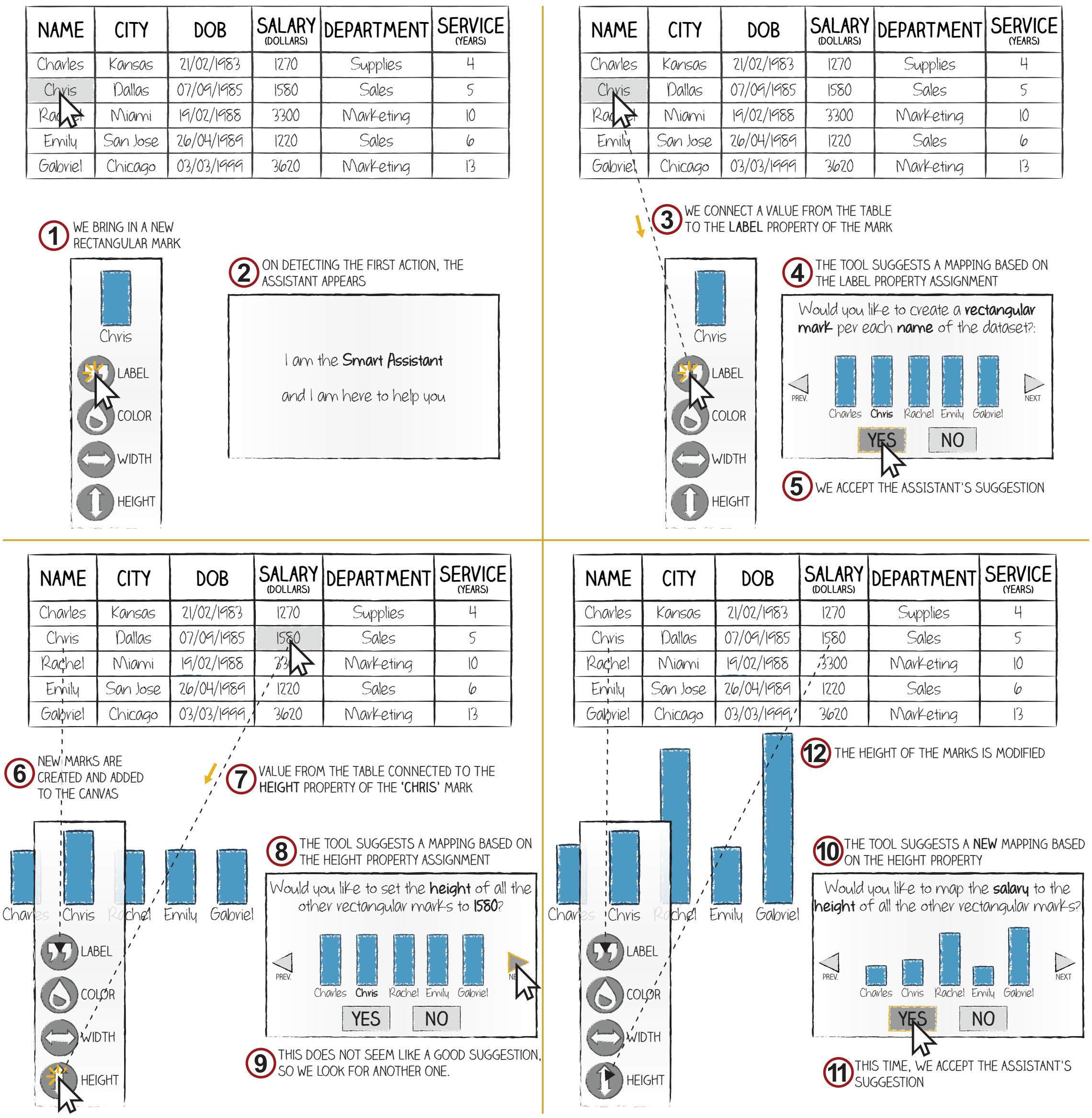


Figure 5.2: In-house Smart Assistant

This solution, shown in Figure 5.2, proposes a visualization process based on a mixed-initiative interaction approach [5]. Visual mappings are specified on a single mark using individual values of the dataset. The tool monitors the designer's mapping actions and provides guidance on how to proceed in further stages of the visualization construction process. As the *Smart Assistant* detects a new mark or modification of a visual property, it infers how the corresponding mapping applies to other data and shows previews of the resulting visualization based on all data values. The user can accept or reject these suggestions, or explore alternatives.

5.3.3 Proxy Mark Widget

In this solution, shown in Figure 5.3, mappings are specified not directly on marks, but through a widget that acts as a *proxy* for a group of marks and provides access to their visual properties. Users specify visual mappings by feeding *sets* of data values to the widget's graphical representations of visual properties. These sets can contain either all the values associated to a data attribute or just some parts. In this design, the former are available through the column headers of the data table while the latter can be manually constructed with a vertical dragging gesture that spans across the desired values. Once all the data mappings have been specified, an output port allows users to drag the resulting visualization to the canvas.

This solution is similar to Tableau's "Marks Card" [196], with the important difference that the user is in full control of the visual properties of marks, including their position in the canvas which can be defined explicitly, rather than implicitly as in Tableau through the "Columns and Rows" shelves [194].

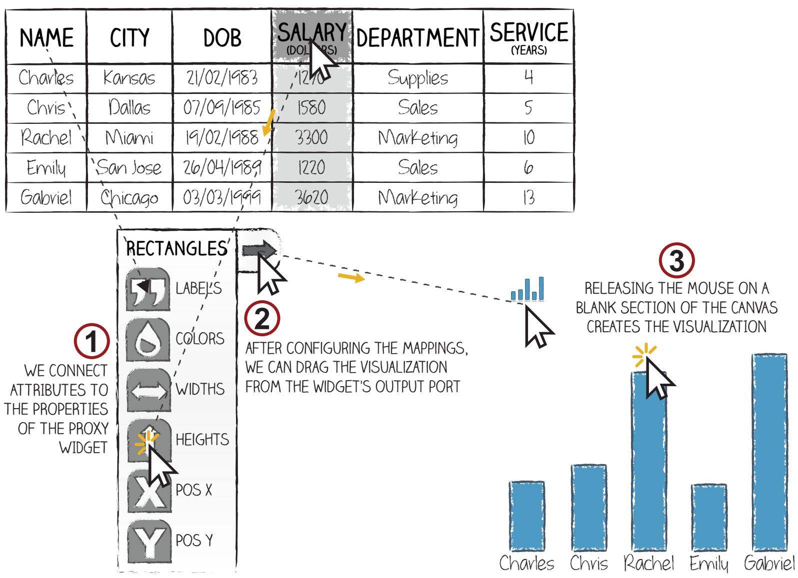


Figure 5.3: In-house Proxy Mark Widget

5.4 Design Workshops

In order to expand and validate the in-house design ideas on facilitating the visualization of larger data sets using iVoLVER, I conducted four design workshops with people of different backgrounds who engaged with the problem for half a day in groups of 3–4. To provide participants with some context, they first worked with Tableau and iVoLVER, before they sketched and discussed their design ideas.

5.4.1 Participants

14 people (8 female, 6 male, 19–49 years old—median 26) from the University of St Andrews participated in the study. Twelve were undergraduate or graduate students in computer science, mathematics, English literature, or museum, art history & gallery studies; one was a practitioner in research communication; and one was a lecturer and researcher in economics & education. None of them had participated in the studies reported in previous chapters of this dissertation.

Expertise in visualization varied from no knowledge (1), experience with visualizations on the web or magazines from a consumer perspective (7), an awareness of visualization concepts, without any practical experience (3), and visualization practitioners with at least one year of experience (3).

Participants' experience with visualization tools and interface design varied. Six did not know Tableau before the study; three had heard of it. Four had used it once, and one was comfortable with it but not proficient. Five participants did not know iVoLVER; five had heard of it. One had used it once, and three were comfortable with it, but not proficient.

Four participants had no experience with designing user interfaces. Nine had varied levels of expertise, ranging from novices (1), basic knowledge from university projects (4), to practitioners with at least one year of experience (4)—their experience ranged from designing mobile or large-display touch interfaces, interfaces for domain specialists, and visualization-based interfaces including Tableau dashboards.

5.4.2 Procedure

Each workshop session consisted of the following stages:

- ***Introduction to Visualization Concepts.*** After filling out a questionnaire about their background (see Appendix E.3), participants watched a 4-minute video explaining the concepts of marks and visual variables and their relevance to visualizing data[1]. Participants kept a printed summary of the video to have at hand during the visualization tasks (see Appendix B.1). This stage ensured that all participants had at least some minimum knowledge on basic visualization concepts.
- ***Tool Introduction.*** A 20-minute session consisted of a live demonstration, introducing participants to the first tool (iVoLVER or Tableau). This included an overview of its interface features, a demonstration of how the tool implemented the visualization concepts, and how to map data to visuals and create visualizations based on a small dataset. Participants were given a printed "tool cheatsheet" to facilitate the subsequent visualization tasks (see Appendices B.2 and B.3).
- ***Visualization Task.*** Participants received one of two datasets and corresponding tasks to complete using the tool at hand. The tool order was counterbalanced across groups, dataset order was counterbalanced across participants within a group. The datasets were the same used in the in-classroom and lab experiments described in Chapter 4: one about oil production (see Table 4.1) and one about movies (see Table 4.2), each containing five data points and five attributes. The tasks included the creation of three different visualizations involving different data attributes. The tasks for both datasets were equivalent in the type and number of attributes. Participants had 30 minutes to complete these tasks. After this, they were introduced to the second tool and used it to complete the visualization tasks with the second dataset.
- ***Group Discussion on Tools.*** The workshop attendees briefly discussed their overall experience with Tableau and iVoLVER, including differences, advantages and limitations.
- ***Design Briefing.*** These hands-on visualization sessions were followed by a briefing highlighting the main goals for the sketching session: To revise iVoLVER as a constructive visualization tool that would: (1) scale to large datasets, (2) support a speedy visualization process, and (3) avoid repetitive interaction steps; all while still supporting transparency and keeping the user in charge of the design decisions.

[1]The same video used in the comparative study described in Chapter 4

- ***Sketching Session.*** Participants then spent approx. 45 minutes sketching design solutions (see Figure 5.4) that would meet these goals. They were advised to revising iVoLVER by either modifying its current features or creating new ones. Participants were provided a sketching toolkit that, among other materials, contained several printed cutouts of a sample dataset (different from the data used in the visualization exercise) that they could glue to A3 sketching sheets and use this as a starting point for sketching a new idea. This configuration also provided a common ground to discuss all participants' solutions later in the workshop. The dataset representations came in small, with five records/rows (see Table 5.1), and large (36 records) versions.
- ***Presentation of Sketches.*** Participants presented their sketches to the group, explaining their ideas toward a better constructive tool.

The workshops ended with a presentation of the in-house solutions leading into a concluding discussion of ideas that had come up in the session.

5.4.3 Data Collection & Analysis

Participants' presentations of sketches were video-recorded (98 min. total), and the sketches themselves were collected (41 pages; available in Appendix E.4). This data forms the basis for the findings reported later in this chapter. The analysis of the video data followed a coding process as proposed by Heath et al. [79] that was informed by observational notes taken throughout the workshops.

Initial coding focused on the visualization stages as addressed by participants and how they envisioned particular features and their implementation. This was followed by grouping the initial codes to classify the participants' solutions based on similarities and higher-level strategies.

Name	City	DoB	Sallary	Department	Service
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13

Table 5.1: Dataset used in the sketching exercise

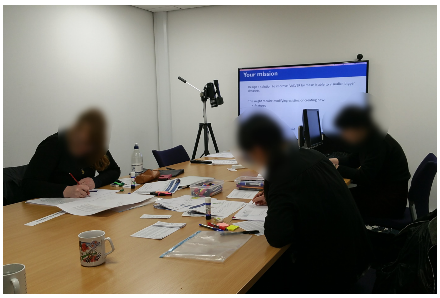
5.5 Results

Participants' approaches to the introductory visualization tasks during the workshops confirmed the observations of the study that compared bottom-up versus top-down visualization authoring using iVoLVER and Tableau Desktop (chapter 4). That is, participants explored more alternative designs when working with Tableau, and produced a more varied set of visualizations when working in iVoLVER. Along the same lines, the group discussion on tools focused on the trade-offs of the two tools' approaches, namely the speed of construction and automation (Tableau) and the transparency and flexibility of the visualization process (iVoLVER). These outcomes validate the setup of the workshops and confirmed that participants understood the problem before the sketching session.

The participants' sketches address three different aspects of the visualization process: (1) data preparation (e.g., formatting, sorting, filtering), (2) interactions with the data (e.g., grouping), and (3) visual mappings specification. The participant's solutions that compose each of these categories are described in the sections below. Note that some of the sketches are annotated in yellow for better comprehension.

(a) Group 1

(b) Group 2



(c) Group 3

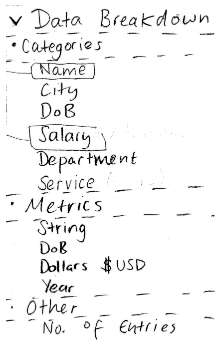
(d) Group 4

Figure 5.4: Sketching exercise setting

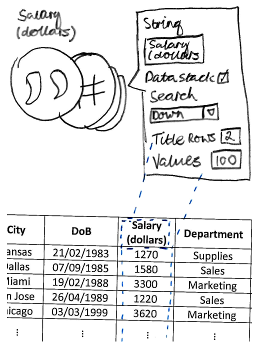
5.5.1 Data Preparation Support

Six sketches (all produced by participants with a computer science background) included solutions to pre-process the data before starting the actual visualization process. Participants with non-technical backgrounds simply assumed that the tool would recognize the data's structure automatically.

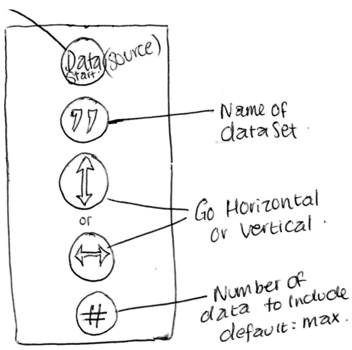
Most of these designs extract and format the data from the table to reflect



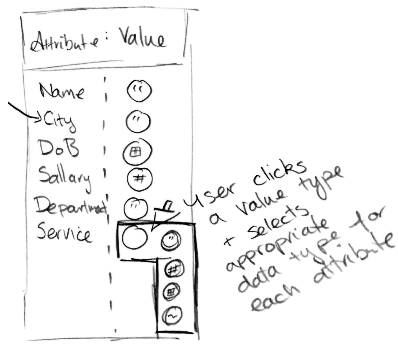
(a) Displaying [P8].



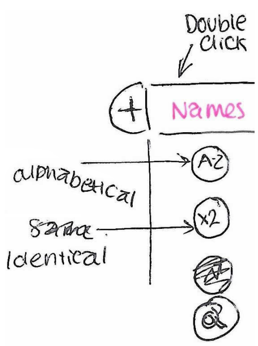
(b) Formatting [P4].



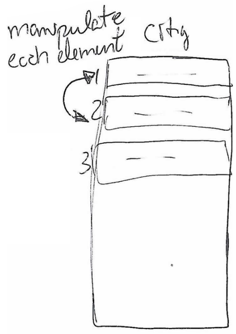
(c) Formatting [P5].



(d) Setting the data type [P2].



(e) Sorting/filtering [P5].



(f) Rearranging [P3].

Figure 5.5: Participant designs that address data-related aspects of the visualization construction process

the relationships between values and attributes. For example, P8 proposed an automatic approach where the tool generates a "data breakdown" pane which organizes the data according to attribute types (see Figure 5.5a). This design resembles Tableau's Data Pane [195] in appearance and functionality.

Other participants proposed a more manual approach for configuring the data extraction process. For example, P4 designed a widget to extract and format the values from an attribute's column (Figure 5.5b) which also allows additional specifications (e.g., such as the number of values to extract). P5 proposed a similar element that extracts and formats values also across columns (Figure 5.5c), and P2 designed a widget to set the data type of each attribute (Figure 5.5d). Other participants designed configurable canvas elements to filter and sort data (Figure 5.5e) and to rearrange the individual values of an attribute (Figure 5.5f).

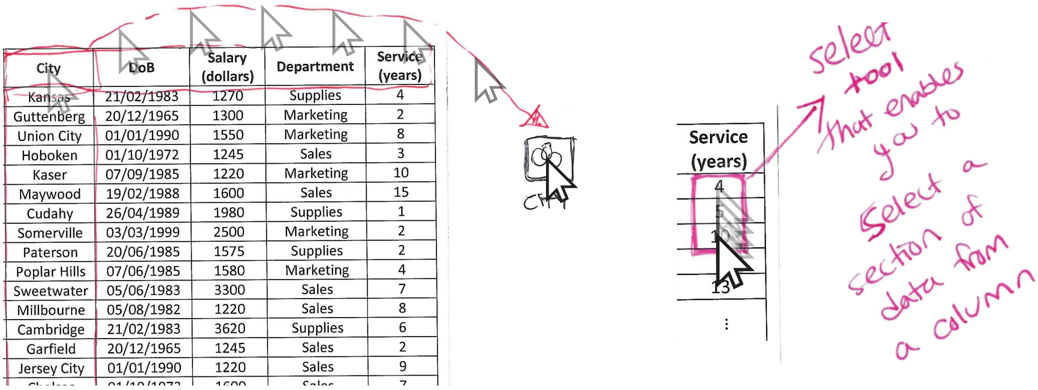
5.5.2 Interactions with the Data

Although some solutions required interaction with individual data values, most participants proposed manipulating several values at a time, highlighting that working with individual values makes the process tedious: *"Both of my ideas are targeted towards improving efficiency because I found that [in iVoLVER] some of the tasks were quite tedious; specially since it's all geared towards single values."* [P8].

11/14 participants proposed accessing all values of an attribute by interacting with the column headers of the data table: *"We want to go from creating data bindings between a single value to some property of an object [a mark]. And we want to go from that to potentially manipulating thousands of objects [marks] at the same time. So, the natural thing to do is to consider the headings, the column headings for the data that we have."* [P7]. An example of interactions with the table's column headers is shown in Figure 5.6a. The creation of value subsets of an attribute was commonly supported via dragging gestures to select the values of interest (Figure 5.6b). Similar to selection of nonadjacent cells in MS Excel, P13 proposed to *"select multiple parts of your data [...] via drag-and-drop or 'control plus click'"*.

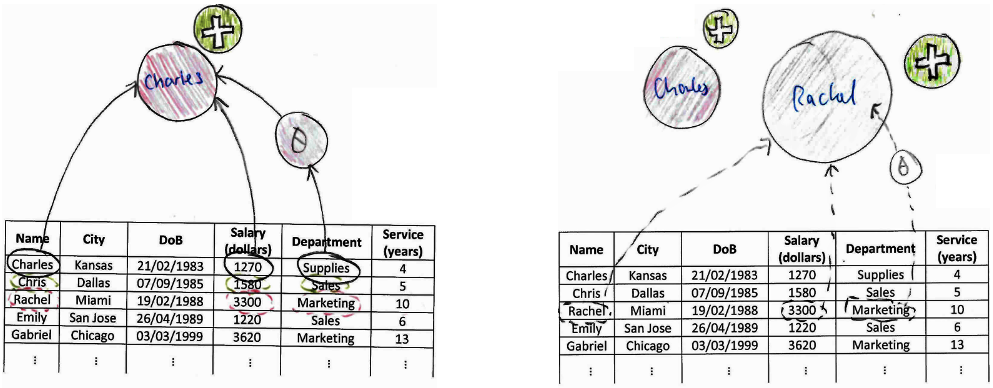
5.5.3 Specifying Visual Mappings

17 design ideas presented by participants focused on how to specify mappings between data and visual properties. These solutions represent six groups of unique ideas that are described below.



(a) With column headers [P3]. **(b)** Direct selection [P12].

Figure 5.6: Interactions with the data table (shown with cursor annotation)



(a) Setting properties of a mark **(b)** Replicated mappings

Figure 5.7: P6's "plus popup" button to replicate visual mappings

5.5.3.1 Mapping Replication

P6 proposed an idea that directly builds on iVoLVER's way of mapping data to visuals, but replicates this mapping process to multiple values and/or attributes. As shown in Figure 5.7a, a data value `Charles` has been dragged into the `label` property of a circular mark, creating a visible link (the visual properties of the mark are not displayed in the sketch). The same is done with a quantitative value of the `Salary` attribute, which links it to the mark's `radius`. Finally, a value of `Supplies` is converted to color through an additional widget before being mapped to the mark's `color` property. Now that the first visual mapping step is completed, the user can press the "plus popup button" to replicate the same process using other data records by clicking rows in the table (Figure 5.7b).

5.5.3.2 Macro Recorder

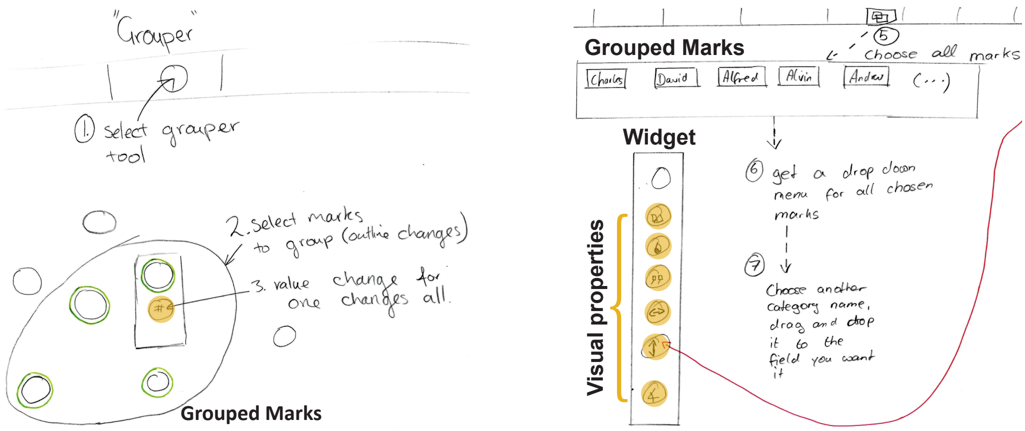
P13's idea was to *"record macros to automate actions"* as described in his presentation: *"You would get one mark and say 'I want to take the country's name for the label and I want to take the height of the bar—if it's a rectangle—for the population of that country'. And you would kind of record your actions. Not as literal as: 'go to this part of the screen with the mouse cursor and click here', but just to say 'now rinse, repeat, wax on, wax off all the way down the table', reapplying the same set of actions as I have."* [P13]

5.5.3.3 Propagation of Individual Mapping Steps

P10's approach (verbally presented) allows users to connect individual data values to visual properties and propagate the underlying mapping to other existing marks on the canvas via a gesture that takes into account the values' and marks' spatial arrangement in the data table and canvas. For example, after connecting a `Salary` value v to the `height` of a rectangular mark M, swiping right on the canvas would modify the `height` of all marks located to the right of M with values from `Salary` that are located below v in the data table. Swiping left would apply the values above v to the marks to the left of M.

5.5.3.4 Bulk Modification of Mark Properties

P2 proposed a selection tool to group marks in the visualization canvas: Figure 5.8a shows a sketch where four circular marks (highlighted in green) have been grouped. Users can access the visual properties of a mark and connect individual data values

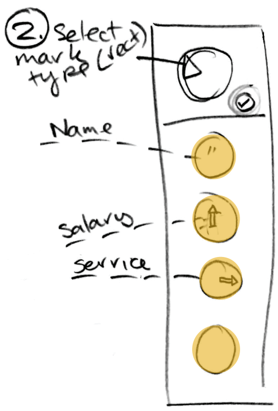


(a) P2 **(b)** P11

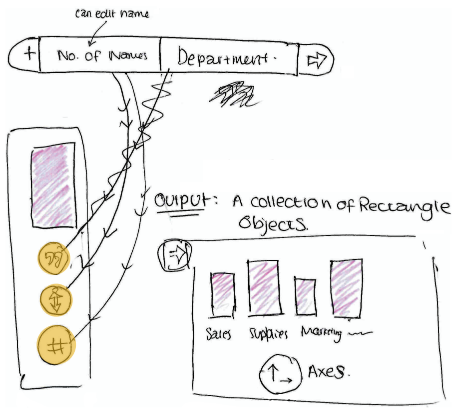
Figure 5.8: Bulk assignation of grouped marks' properties

to it, which directly affects the other marks: *"If you change [a visual property] for one thing [mark] in that group, that [property] changes in the same way for all of them."* [P2]. This operation can make use of actual data, but it can also facilitate changing a particular visual property across all marks: *"It's not intended for large datasets where the attributes need to be different for every one. It's intended for, like, if you wanna change the color of several marks at the same time."* [P2].

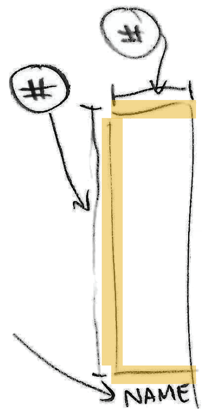
P11 proposed an idea that also involves a group selection tool (Figure 5.8b). Here, the tool automatically associates a widget to any user-created group of marks. Double-clicking a group evokes this widget which provides access to the marks' visual properties. Sets of data values can be connected to each of these properties. When receiving a new data link, the corresponding property of each mark within a group changes.



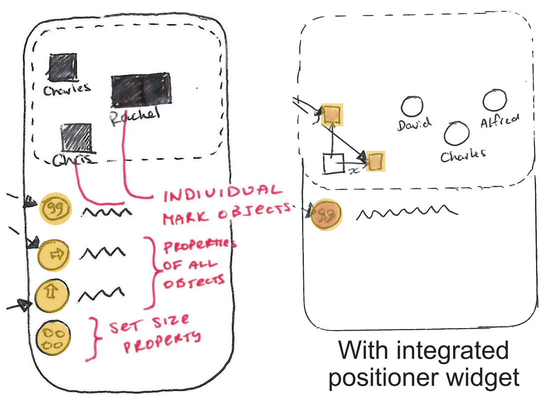
(a) P2



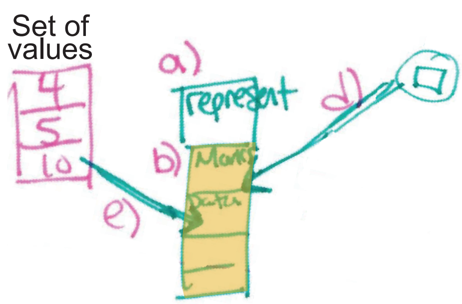
(b) P5



(c) P6



(d) P7



(e) P12

Figure 5.9: Designs that specify visual mappings through a proxy object. The yellow highlighting shows the proxy's properties where data is connected.

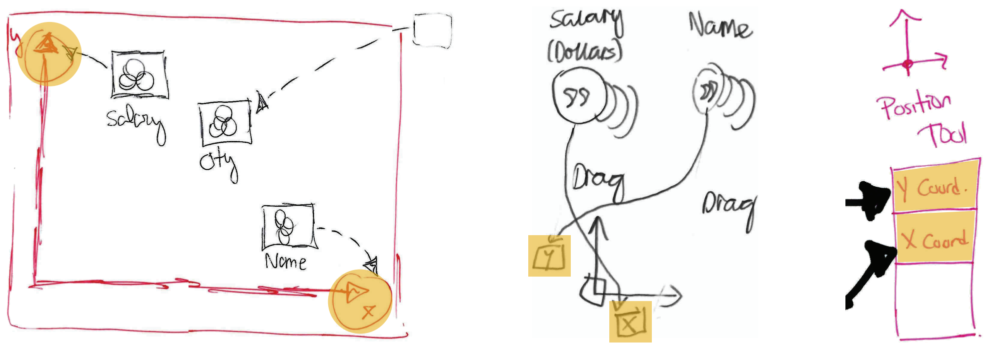
5.5.3.5 Collective Proxy Objects

Five participants (P2, 5, 6, 7, and 12) proposed specifying the mappings of several marks using a single graphical object (Figure 5.9). The circular object in Figure 5.9a, for example, is not an individual visual mark, but a widget that provides a list of four visual properties. The sketch shows the data attributes `Name`, `Salary`, and `Service` connected to the widget's `label`, `height` and `width` visual properties, respectively. The widget eventually generates the visualization by instantiating a group of marks, each representing the specified mappings. I refer to these designs as *collective proxy objects* as they support a visualization process that involves an object that acts as a "proxy" for a collection of marks. Participants' ideas proposing this type of solution differed in the following aspects:

- ***Shape property.*** Three participants sketched ideas where the visual appearance of the proxy object determines the shape of the marks it generates. For example, a rectangular proxy object creates a bar chart like visualization (see Figure 5.9b). P2 and P6's ideas (shown in Figure 5.9a and 5.9c) are also based on a predefined shape property. To overcome the inherent limitation of this solution that would only allow to construct visualizations whose marks have the same shape, P7 proposed a *shapeless* proxy object with an additional `shape` property to be specified for each mark independently. P12 proposed a similar idea (see Figure 5.9e; here, the shape property is called "mark").

- ***Time and space of rendering.*** Four proposed ideas require the user to explicitly trigger the generation of the visualization once the visual mappings are configured. For example, in P2's design (Figure 5.9a), this is done via the button located at the bottom-right of the circular proxy object. Other participants proposed rendering the visualization via a drag-and-drop gesture from the proxy object onto the canvas. Only one participant proposed a design that does not require a rendering step but allows changes on marks to take place in real time, as data is connected to the proxy object's visual properties. As shown in Figure 5.9d (left) marks (the three black rectangles) reside within the proxy object itself, which also acts as a container for the resulting visualization.

- ***Position mappings.*** The proxy objects of four participants lack location properties, but implement position mappings as in iVoLVER: by linking marks to an external *positioner* widget (i.e., a locator). In contrast, P7 incorporated the locator within his proxy object design (Figure 5.9d, right).



(a) P3 **(b)** P4 **(c)** P12

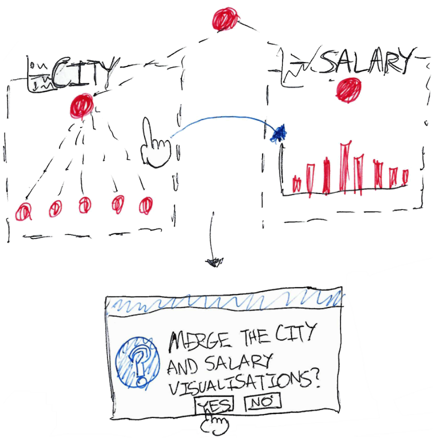
Figure 5.10: Axis-based solutions. The yellow highlighting indicates the axes' points where data can be connected.

5.5.3.6 Axis-based Solutions

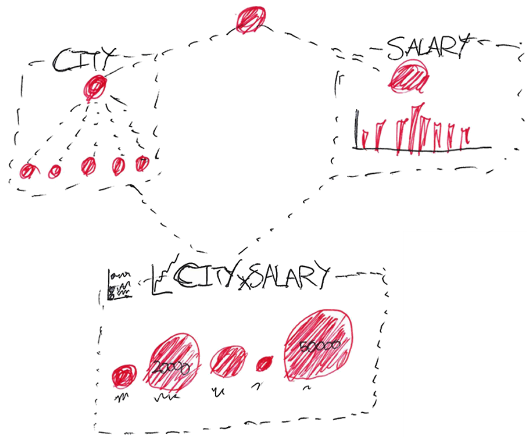
Three participants proposed ideas that include axes as a fundamental visualization component. These widgets enable the specification of position mappings when sets of quantitative values are connected to the widget's axes (Figure 5.10). When the connected data contains categorical values, the widget applies heuristics to determine other types of visual encodings. For example, when connecting a set of quantitative values to one axis and a set of categorical values to the other, the widget generates a bar chart. The widget therefore decides the marks that compose the visualization, i.e., the data input and the order in which it is connected determines the type and layout of the visualization. This idea is comparable to Tableau's "Columns and Rows" system where marks and visual properties are determined by the data attributes on each shelf.

5.5.3.7 Simple Representations as Building Blocks

P1 proposed an idea where the tool decides the visualization design based on the data attributes types. For example, Figure 5.11a (top row) shows two visualizations automatically generated by the tool for two data attributes: *"[the tool] knows it [the* `city` *attribute] is categorical, so it makes up something like this. For the* `salary`*, it creates a distribution [histogram] of the actual values themselves."* [P1]. To define more complex visualizations, the user can merge simple representations via drag-and-drop gestures (see Figure 5.11b, bottom, showing *"the average salary of every city"* [P1]). In this solution, all visualization design decisions are made by the tool based on the data attributes. The more attributes, the more complex the decisions inferred by the tool. Although not sketched, P1 said that *"there will be ways to sort of*



(a) Individual views



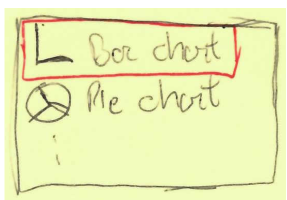
(b) Resulting Visualization

Figure 5.11: Merging simple views to create more complex one [P1]

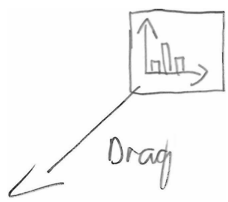
customize this [resulting visualization], 'cause you are gonna want to customize them in different ways, presumably. But it [the tool] will give you something for a start. It will give you something basic to work with and then you can carry that on."

5.5.3.8 Chart Galleries

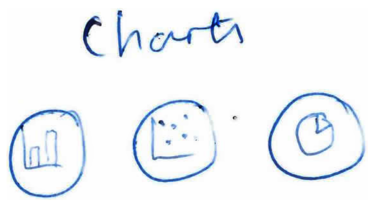
P3, 4 and 13 proposed in their designs a gallery of charts for the user to select in order to represent data attributes (Figure 5.12). P13 explained that this part of his design was similar to MS Excel's charts gallery or Tableau's "Show Me" pane. The galleries from P3 and P4 additionally provide access to other type of designs. For example, dragging a bar chart from P3's gallery onto the canvas creates an axis-based widget.



(a) P3



(b) P4



(c) P13

Figure 5.12: Sketched chart galleries.

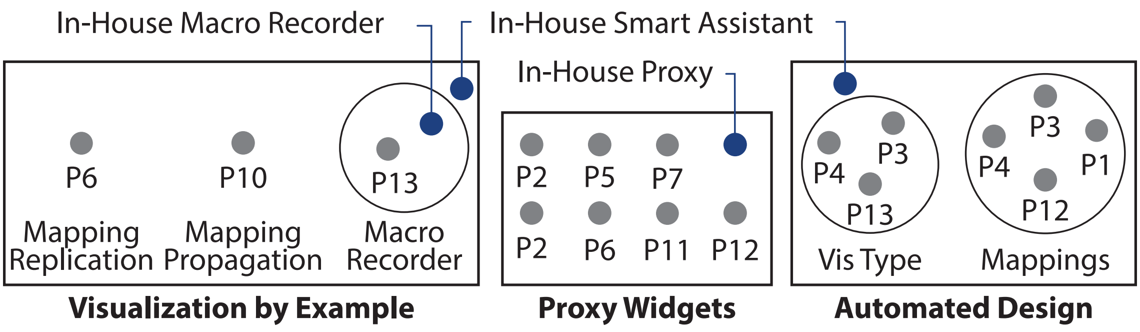


Figure 5.13: Families of visual mapping strategies. Gray circles represent the participants' designs; in-house solutions appear in blue.

5.6 Families of Design Solutions

The participants' design ideas presented above and the in-house solutions reveal three different design families (see Figure 5.13): (1) visualization by example, (2) collective proxy objects, and (3) automated principled design. The sections below characterize these families alongside their underlying design strategies: (1) automated iteration, (2) abstraction, and (3) automated choices.

5.6.1 Visualization-by-Example: Automated Iteration

Four of the presented design ideas first manually specify visual mappings based on a single data instance (e.g., one record or table row) which are then automatically applied using other data or the entire set (Figure 5.13, left). The "plus popup button" proposed by P6 (Figure 5.7) and P10's gesture-based mapping propagation idea represent this visualization-by-example approach. This family also includes P13's description of a *Macro Recorder*, the in-house one (Figure 5.1), and comparable macro recording solutions in commercial tools such as MS Excel.

These solutions directly address the issue of having to repeatedly specify the mapping between individual data points and visual elements that arises in iVoLVER due to its bottom-up approach to visualization authoring. To this end, visualization-by-example solutions implement *automated iteration* as a strategy that allows the application of design decisions reflected by an individual data-driven visual element to a larger scale.

As shown above, implementations of this strategy differ in terms of *how* automated iterations actions are specified and *when* results become visible. For example, P6's "plus button" idea specifies how a data instance is represented by a mark's visual properties. In contrast, macro recorder solutions specify mappings as ordered sequences of actions carried out within a particular temporal interval. Both solutions apply mappings in bulk, while others, such as P10's gesture-based mapping propagation, do this as new individual mappings steps are specified, showing how the automated iteration is progressively carried out across the data.

5.6.2 Collective Proxy Objects: Abstraction

Eight sketches implement interface elements (e.g., a widget), separate from the resulting visualization, to specify visual mappings between data and visual elements (Figure 5.13, middle). Such proxy objects can take incoming connections with individual values, sets, or data attributes. Modifications to the visual mapping happen through the proxy object and propagate to all the corresponding visualization elements. The design solutions in Figures 5.3, 5.8b, and 5.9 show different implementations of collective proxy objects. Other examples include Tableau's "Marks Card" [196] which acts as a proxy of the final visualization. P2 presents a different implementation of this design approach where every visualization element is also a proxy for all other elements within its group (Figure 5.8a).

Design solutions in this family are based on *abstraction* as a design strategy: a single interface element represents—logically—multiple data items and the visual properties of the corresponding marks while the individual constituents are hidden. Abstraction is a general term well discussed in psychology, semiotics and other areas, but here it best conveys this strategy's meaning. As a design strategy to scale constructive visualizations, abstraction affects access to visual properties and mapping options, and to the data itself: high degree of abstraction may allow visual mappings at an attribute level, while a low degree will enable mappings at the level of individual values.

5.6.3 Automated Principled Design: Automated Choices

Seven of the presented designs support a top-down approach to visualization construction (in the sense defined in Chapter 4), where visual mappings are decided and automatically applied by the visualization tool based on pre-defined principles as specified in its implementation and not by the visualization designer (Figure 5.13, right).

For example, in P4's solution (Figure 5.10b) the designer selects the data attributes and loosely specifies the visualization type by mapping each to the chart's axes. The tool then specifies the chart type that fits these data types. Similarly, in P1's design, two visualizations are automatically merged into one, whose design is decided based on the initial data types involved (Figure 5.11). In P13's solution the tool offers a variety of chart types which will fit the attribute types selected by the designer. Once the designer selects the chart type, the tool applies the necessary visual mappings. This is similar to how MS Excel's chart gallery [57] or Tableau's "Show Me" [126] pane operate. P3's solution (Figure 5.10a) supports scenarios where the designer chooses a visualization type (e.g., a bar chart), but then maps data attributes of an unsuitable type (e.g., two continuous attributes): the tool automatically performs operations to make the data fit the visualization type (e.g., bin or average data values).

The underlying strategy of this family of designs is that of *automated choices* in which explicit decisions by the designer are minimized, and the tool makes choices such as: (1) the visualization type (e.g., the type of marks and visual attributes), (2) the mappings between data and visual variables, and (3) applying data transformations (e.g., binning a continuous data attribute to map to a discrete visual variable). While this strategy leaves many design decisions to the tool, automated choices can be implemented as mere suggestions or as configurable by the designer.

The three strategies to scale constructive visualization approaches to large data sets—automated iteration, abstraction, and automated choices—are not necessarily mutually exclusive; design implementations can apply multiple strategies. For example, the in-house "Smart Assistant" solution (Figure 5.2) can be considered as an implementation of the automated choices strategy, as it makes unprompted decisions about generalizing visual mappings. But, it also implements the automated iteration strategy as it allows the designer to explicitly apply defined visual mappings to additional data points.

5.7 Agency and Atomicity: A Design Space for Visualization Tools

The families described in the previous section are clusters of design solutions based on strategies to implement constructive visualization tools that scale to larger datasets. This section presents a design space emerging from these strategies

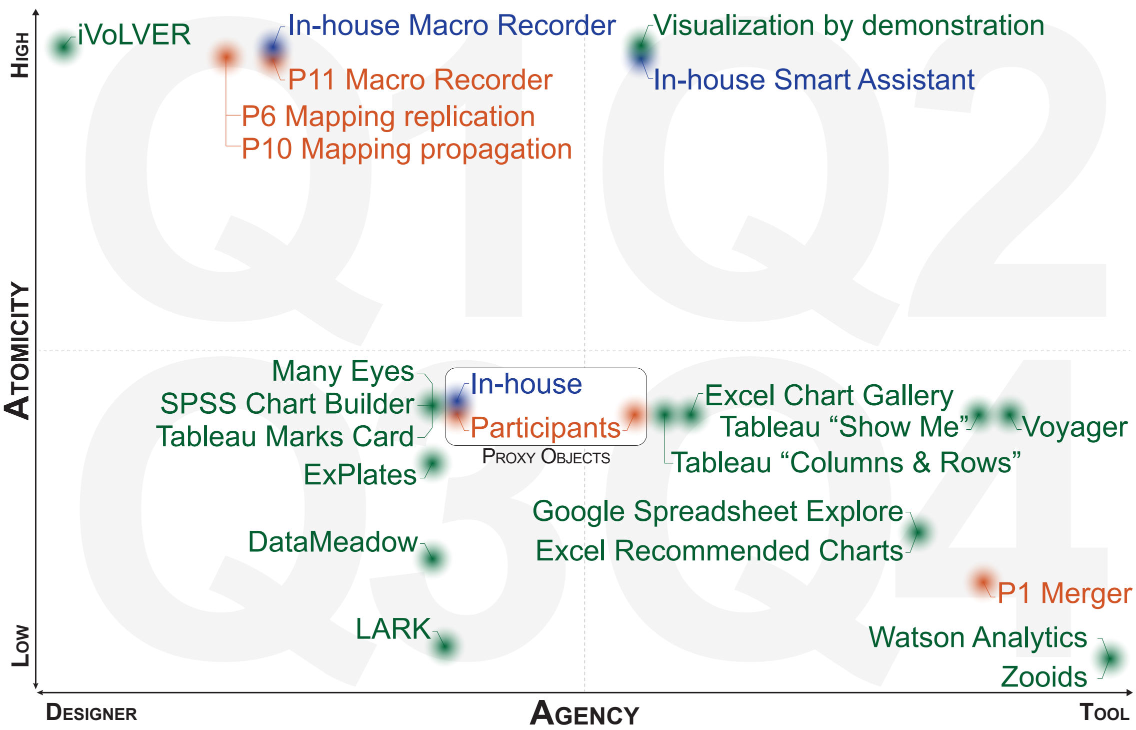


Figure 5.14: Design space defined by agency & atomicity with existing tools (green), including MS Excel [57], the Google Sheets Explorer feature [68], and SPSS Chart Builder [98], alongside in-house (blue) and workshop solutions (red). The workshop solutions from the "collective proxy objects" family span horizontally because they combine abstraction with different levels of automated choices.

which is defined by *agency* and *atomicity* as two qualitative dimensions. This design space summarizes the differences between the proposed solutions (both in-house and from the workshops) and places some existing representative research and commercial tools in relation to the solutions of the design study.

5.7.1 Agency

In this design space, agency refers to *who* carries out the visualization process: the tool or the human designer. A visualization tool can make a range of decisions during the design process, such as selecting the visualization type suitable for a designer-selected data subset (Figure 5.12c [P13]), or even performing actions on behalf of the designer, such as applying a mapping repetitively to the full dataset (Figure 5.10b [P4]) or just to the next record (Figure 5.7 [P6]).

This dimension is represented as the horizontal axis in the design space diagram shown in Figure 5.14. Design solutions in which more agency is placed on the visualization tool, that is, where the tool is in charge of important design decisions and actions (e.g., visual mappings), are located more to the right. To the left on this axis are purely constructive tools, such as iVoLVER.[2]

Agency is not exclusively linked to the automatic choices strategy. Abstraction and automated iteration also place some agency on the tool, as it takes over designer activities.

5.7.2 Atomicity

Atomicity refers to the level at which the tool allows the manipulation of both data and visual representations; that is, *what* designers can see and manipulate. At the most atomic level of data the tool represents and allows the manipulation of individual values (e.g., a single number in a table cell). Similarly, at the most atomic level a designer can manipulate individual marks (e.g., one bar of a bar chart). This is how iVoLVER operates. Designs with low atomicity operate mostly on groups of values, at an attribute level, or on groups of attributes (or on groups of marks, or even groups of small multiples on the visual side). Most solutions of the "collective proxy objects" family specify the visual mappings of many marks through a single widget that takes data attributes—not individual values—as an input (e.g., Figure 5.9a [P2]).

This dimension is represented as the vertical axis in the design space diagram of Figure 5.14. Solutions higher in this axis have the highest atomicity (e.g., iVoLVER, which is based on the manipulation of individual values and marks); solutions lower down represent data mostly at an attribute level (e.g, the data column becomes a single element to manipulate in Tableau, Many Eyes [213] and the designs of Figure 5.9). In DataMeadow [56], the visualization process involves manipulating data attributes, but the entire dataset can also be represented and manipulated through a single visual element. Lark [205] supports an even more lower atomicity: a complete dataset is represented as a single visual element that moves through different transformation steps and the visualization's components cannot be manipulated directly.

[2] Note that even a tool like iVoLVER, built from the ground up with a constructive philosophy, makes some decisions for the designer. For example, if the designer drops a data value on top of a mark, iVoLVER will automatically assign it to a default visual property.

5.7.3 The Design Space: Strategies, Quadrants & Limitations

The set of strategies described in the *Families of Design Solutions* section can also be described as a collection of design tropes that allow designers of visualization tools to explore and discuss important options and their corresponding consequences. For example, adding automatic iteration to a purely constructive tool will place more agency on the tool: now the tool does something for the designer. Adding abstraction decreases the level of atomicity (abstraction hides access to the more atomic elements) and also places agency on the tool: as with automatic iterations, the tool applies operations to all individual data values and visuals on behalf of the designer. Finally, implementing automated choices shifts the agency balance to the tool side; adding automated decisions to any tool will move it closer to the right end of the design space.

Besides the in-house (orange) and workshop (blue) design solutions, the design space of Figure 5.14 shows a a selection of representative existing visualization tools (green). Several interesting observations can be derived by looking at the distribution of the tools in the quadrants of this space. First, most of the surveyed tools (green) gravitate towards the bottom right corner of the space (Q_4), with the exception of iVoLVER. Second, the right side of Q_1 is populated mostly by solutions from the workshop (red). Third, there are parts of the space that are barely populated, most prominently Q_2, which represents tools that make many decisions on behalf of the designer but stay at a high atomicity level. I believe that there is potential for tools in these underpopulated areas. For example, a Q_2 tool could be a "visualization tutor" that walks a learner step-by-step through the visual mapping process using individual data values.

The presented design space should be understood as a qualitative tool as opposed to an exact quantitative metric to specify tool differences. Therefore the positioning of tools and solutions in Figure 5.14 is open to interpretation and is meant to promote discussion on how agency and atomicity are addressed within the current visualization tools landscape. Some tools also appear in several places because they incorporate multiple strategies to create visualizations (e.g., Tableau's "Show Me" [126] pane and "Columns and Rows" [194]). Similarly, the workshop solutions grouped under the "collective proxy objects" family span horizontally because they combine abstraction with different levels of automated choices.

5.8 Discussion

This section presents an interpretation of the results, explains the relevance of the work described in this chapter, discusses its implications, and provides a critical look at the current landscape of visualization tools.

5.8.1 The Agency-Atomicity Design Space

The design explorations described in this chapter generate a design space based on agency and atomicity that conceptualizes the challenge of scale in visualization tools from the perspective of making visual mapping processes transparent and promoting informed and critical hands-on exploration.

This design space can be seen as a lens to look at and discuss existing design solutions and strategies and to inspire new ones. It brings constructive approaches into view in relation to more conventional tools, offering designers a catalog of examples and a deeper understanding of available solutions and their trade-offs. It might also help the visualization community to identify promising unexplored spaces. I believe that this space complements and further refines other characterizations of the existing landscape of visualization tools, such as Tory and Möller's taxonomy [206] and the top-down vs. bottom-up continuum to visualization authoring presented in Chapter 4.

The design space here proposed also allows to characterize visualization processes achieved with digital tangible tools. *Zooids* [117], robots that can behave as physical pixels, have b0een used in data physicalization. In these scenarios, the visualization is triggered by the designer but the agency of the design process itself resides entirely on the zooids. The robots arrange themselves autonomously to compose the visualization in the physical world. Atomic pieces of data (e.g., individual values, attributes) are not accessible to the designer. Instead, the dataset is represented as a whole in a process in which the visualization designer is mainly an spectator of decisions taken by the tool. Thus, the tangible visualization scenario described in [117] is located in Q_4.

5.8.2 Building Better Visualization Tools

The overarching goal of the work presented in this chapter is to help create better visualization tools. This means designing tools that are able to deal with datasets that are larger than just a few records (scalable), that are not tedious but inspire critical exploration, that are transparent and understandable, and that support mindful design, as well as control and ownership of the results by the designer.

All design strategies presented here help to deal with the scalability issue, but not all of them address the other desirable outcomes listed above. The findings presented in Chapter 4 indicate that tools located far from the top-left corner of the design space can, while increasing scalability, have a negative impact on other desirable characteristics of use. The next step is therefore to find the "sweet spot" that provide maximum advantages. Although the work presented in this chapter supports this goal, further research is needed to investigate how different levels of agency and atomicity affect visualization tool use. Moreover, where the sweet spot is will depend on the particular task at hand (e.g., exploring new vs. familiar data), the audience (e.g., users' levels of expertise), and the usage scenario (exploration vs. presentation).

5.8.3 A Critical Look at the Current Landscape of Visualization Tools

Placing existing visualization tools into the presented design space defined by agency and atomicity (see Figure 5.14) suggests that much effort has gone into creating tools that support the rapid design of visualizations through a low level of atomicity and, arguably, little effort from the designer (Q_4). Such visualization tools are highly valuable and are indeed widely used as part of visualization practitioners' design, research, and teaching practice. However, while supporting rapid visualization creation in this way (through pre-defined or automatically inferred design choices) has its place, there is a need for more alternatives.

Recent work on supporting more manual constructive visualization processes and physicalization [94, 96] is already going in this direction, but there is a urgency for more approaches like this. Data analysis and required literacy [26] is starting to have an effect beyond professional scenarios on our personal lives [88, 204] and even on the youngest members of society [6, 134].

A visualization is not just a means to an end. Reflection on the data and insight generation take place also *during* the process of constructing the visualization itself [84, 217]. Supporting a fast visualization process is therefore not necessarily more "efficient" or beneficial as it may gloss over important details of the data, patterns, uncertainties and open questions, and opportunities for exploration and critique. Previous research from (data) design practice has shown that active involvement [148], and repetition [15] can foster creativity and critical thinking. When visualization is considered as an active data exploration process, the visualization tool itself might heavily influence insights and reflections. But even if we consider visualization as something to be consumed rather than actively explored and manipulated in a hands-on way, the visualization tool plays an important role.

There is a rhetorical nature to visualizations [92], where choices in visual representation and composition steer people's thinking of the presented topics. Tools influence the design process of visualizations and, through their implementations, can make decisions on behalf of the designer. Their power to influence messages "by design" should not be underestimated. Visualization tool designers have a responsibility, and I hope that the design space presented in this chapter fuels a discussion in the community and inspires new and alternative avenues to visualization tool design. In fact, linking these findings to philosophy and cognitive science, in particular, cognitive embodiment [43], media studies [133], feminist theory [199] and, most relevantly, the study of agency and technology [116, 199] may lead to even more research avenues.

5.9 Limitations and Questions for Future Work

Although the conceptualization of the problem, catalog of examples, strategies and the design space presented in this chapter advance the understanding in how to design scalable digital constructive visualization tools, there is much left to do, especially characterizing the effects of the different design strategies on the outcomes that we care about. Perhaps most interesting for future research is to investigate how different design strategies presented here impact the visualization design process, in particular regarding mindful and critical design, transparency and authorship.

It is also important to highlight that the outcomes of the design workshops should not be interpreted as quantitative evidence of what strategy or type of tool is best understood or comes most "naturally" to participants. The workshops were designed to find as many divergent solutions as possible. That is, to expose the nature of the design space, rather than identifying a "best" solution. They were not meant to serve as a faithful characterization of what people want or understand.

Although the workshops included people with different backgrounds and levels of experiences, they are also limited in multiple ways: (1) participants are influenced by the tools they know; (2) participants might be influenced by the tools that were used to train them and contextualize the problem; and (3) participants might need more time to provide a wider variety of designs.

5.10 Conclusion

This chapter explored the design space of visualization tools that could combine the benefits of constructive or bottom-up visualization tools, namely transparency and active involvement in the visualization process with those of top-down approaches, such as rapid visualization creation and applicability to large datasets.

The analysis included solutions resulting from an in-house design exercise, and from a group of participants of four design workshops with a varied range of skill sets and levels of expertise in visualization. This data led to a catalog of solutions, a set of three design strategies (automated iteration, abstraction, and automated choices) and a conceptualization of the design space based on agency and atomicity. Finally, based on the conducted analysis, this chapter also provides a critical discussion of the current landscape of visualization tools that hopefully will help tool designers and the InfoVis community in general to consider better options in the design of visualization tools.

Chapter Six

A revised iVoLVER

Everything should be made as simple as possible, but not simpler.

— Albert Einstein (attributed)

The previous chapter presented the results of a design study that uncovered a set of generic strategies to overcome the scalability problem of ConstructiveVis (described in section3.8.2). In this chapter, I consider these strategies to present the design of a more scalable version of iVoLVER that incorporates abstraction, automated repetition, and automated choices to allow for the visualization of larger datasets. This chapter also discusses the implications that applying these strategies would have on the specific implementation of iVoLVER. Rather than providing the ultimate constructive visualization tool, this chapter seeks to explore how the findings of Chapter 5 can be applied to an actual tool implementation that, in this particular case, is a visual programming environment.

6.1 Goals

In this chapter, I set out to revise iVoLVER's original design to achieve a tool that:

- ***G1:*** Promotes a visualization process in which the designer (not the tool) is in charge of the designs decisions (e.g., visual mappings);
- ***G2:*** Allows visualization of larger datasets without requiring excessive amounts of effort, time, and repetitive steps from the user; and
- ***G3:*** Applies the designer's decisions in a transparent fashion.

These goals are motivated by the findings of Study 2, reported in Chapter 4. I seek to preserve the benefits that iVoLVER's constructive approach brings into non-experts' visualization process in a tool that can be used in analytical settings that involve larger datasets. Although other tool characteristics would be as desirable as the ones listed above (e.g., support to work with multiple automatically linked datasets, support for non-tabular data), I prioritize here goals that are within the scope of this dissertation.

The tool this chapter presents is mainly informed by the outcomes of the design study presented in Chapter 5. It is important to acknowledge, however, that the design here proposed is one of many possible alternatives. Other solutions could achieve the pursued goals in different ways.

6.2 Revisions

To achieve the goals listed in the previous section, the revised iVoLVER supports different degrees of abstraction at both the data and visual representation levels. It also modifies how data tables are drawn in the iVoLVER canvas to enable the representation of bigger datasets without overcrowding the canvas. This version of the tool uses a mark design that borrows inspiration from the "collective proxy objects" (discussed in section 5.6.2 of Chapter 5). By allowing marks' visual properties to take either individual data values or sets of these, proxies avoid the execution of repetitive interaction steps. Together, these modifications increase the overall scalability of the constructive approach to visualization that iVoLVER promotes.

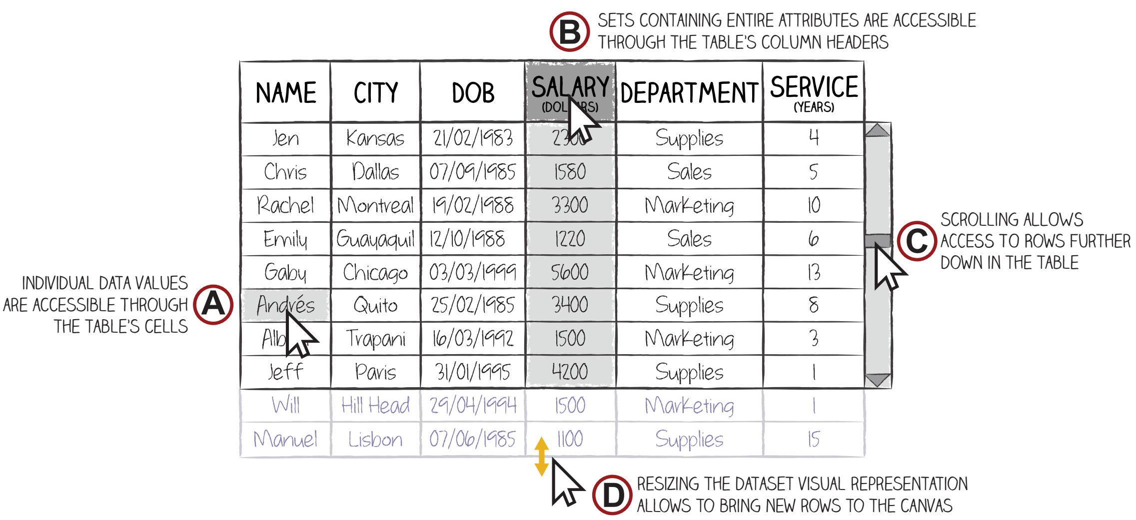


Figure 6.1: Visual representation of datasets in the revised iVoLVER

6.2.1 Visual Representation of Datasets

Besides the need of repetitive interaction steps, working with larger datasets in iVoLVER can be problematic due to the amount of screen space they require when drawn on the canvas. This problem is a direct consequence of iVoLVER's "avoiding hidden elements" principle, by which all the records of a dataset are available on the tool's canvas at all times. Representing entire big datasets also imposes the need of lengthy dragging gestures and significant numbers of zooming and panning operations.

Figure 6.1 shows the modified visual representation of datasets on the iVoLVER canvas. This revised design enhances the scalability of the tool by: (1) allowing access to both individual data values (A) and entire attributes (B); (2) rendering only a few of the first records of the dataset and allowing access to others via scrolling (C); and (3) by making the canvas object resizable to make rows further down permanently visible only when required by the designer (D).

NAME	CITY	DOB	SALARY (DOLLARS)	DEPARTMENT	SERVICE (YEARS)
Jen	Kansas	21/02/1983	2300	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Montreal	19/02/1988	3300	Marketing	10
Emily	Guayaquil	12/10/1988	1220	Sales	6
Gaby	Chicago	03/03/1999	5600	Marketing	13
Andrés	Quito	25/02/1985	3400	Supplies	8
Albert	Trapani	16/03/1992	1500	Marketing	3
Jeff	Paris	31/01/1995	4200	Supplies	1
Will	Hill Head	29/04/1994	1500	Marketing	1
Manuel	Lisbon	07/06/1985	1100	Supplies	15

(a) Full dataset (showing 10 records).

NAME	CITY	DOB	SALARY (DOLLARS)	DEPARTMENT	SERVICE (YEARS)
Jen	Kansas	21/02/1983	2300	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Montreal	19/02/1988	3300	Marketing	10
Emily	Guayaquil	26/04/1989	1220	Sales	6
Gaby	Chicago	03/03/1999	5600	Marketing	13

(b) Resized table showing only the first five records of the dataset.

Figure 6.2: Revised canvas representation for datasets on the iVoLVER canvas. These visual representations can be resized to show full (a) or partial views of the data contained in the dataset.

These modifications can potentially impact iVoLVER's avoidance of hidden elements, but are essential to overcome some of the issues that lead to the scalability problem of ConstructiveVis in the tool. The design allows, nevertheless, to fully represent full datasets of moderate size within the canvas. Figure 6.2a shows an example of this for a 10-record dataset. In the depicted sketch, the scroll bar located to the right of the dataset appears disabled as no rows are hidden (the table has been resized to its maximum vertical size).

Figure 6.2b shows a smaller visual representation of the same 10-record dataset. In this case, the table has been resized to show only its first five records. The scroll bar of the right appears now enabled and the three lines at the bottom of the dataset visual representation indicate that some data records are hidden and that they are available through resizing or scrolling. This revised visual representation allows working with datasets regardless of the number of records they have.

The remaining of this chapter uses the dataset of Figure 6.2 to illustrate how the revised iVoLVER supports a scalable constructive visualization process.

6.2.2 Revised Mark Design

As mentioned in section 5.5.3.5, proxy objects with a fixed, non-configurable `shape` property only allow to construct visualizations whose marks have all the same shape (e.g., only circles, only rectangles). To overcome this limitation, the new iVoLVER mark design includes a `shape` property. In turn, this has implications on how the mark's size is controlled.

In the original version of iVoLVER the size-related properties of a specific mark depended on its `shape` property. For example, circular marks had a `radius`, squared marks had a `side` property, and rectangular ones had a `width` and a `height`. This strategy would not work for a mark design whose `shape` property can take sets of shape values (e.g., the set [`circle`, `rectangle`, `circle`]). In the revised design, the size of marks is always determined by `width` and `height` properties, regardless of their `shape`. This also avoids the need to change the set of properties that individual marks have depending on their shapes.

6.2.3 Specifying Visual Mappings

In Figure 6.3 a value from the `Name` column and another from the `Salary` are connected to the mark's `label` and `height` properties, respectively (steps 2 and 3).

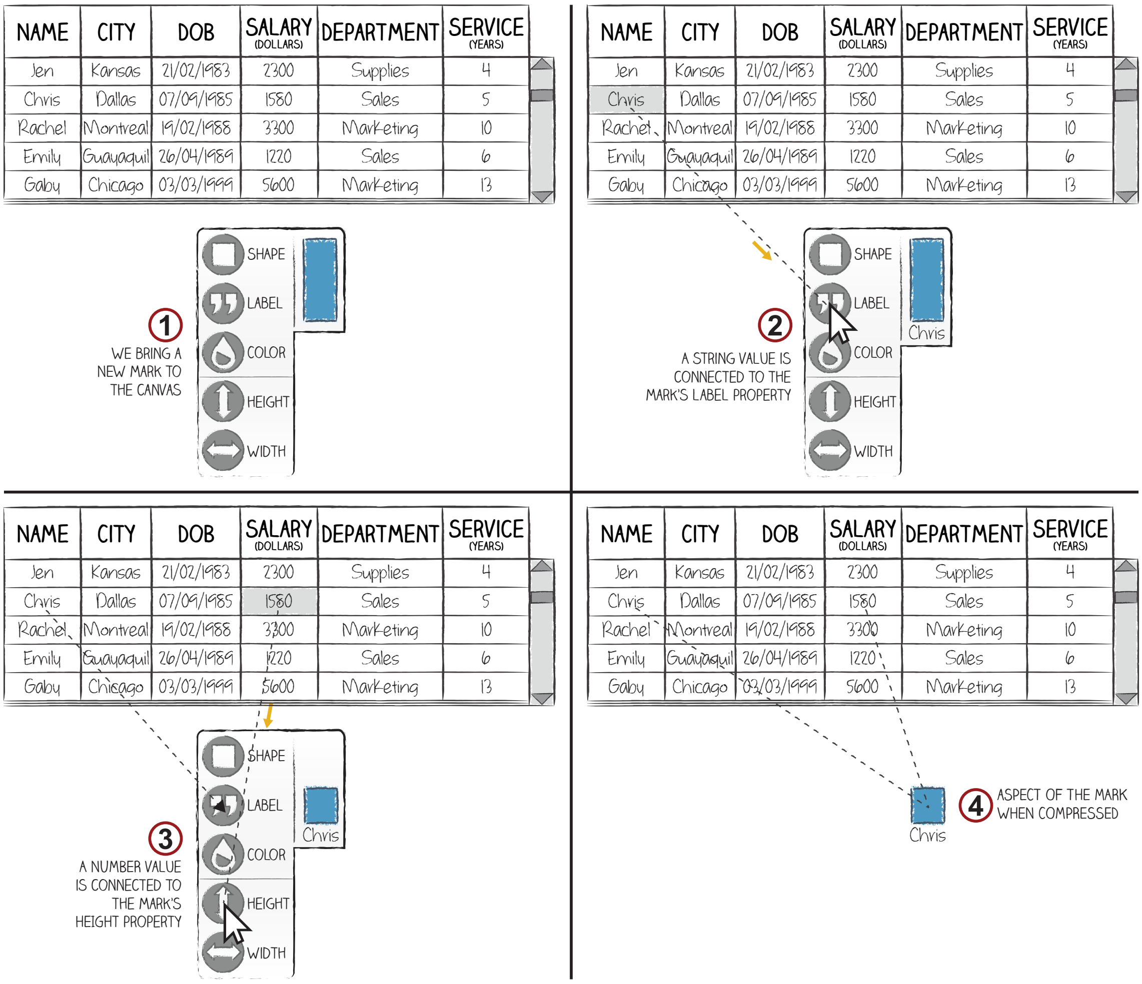


Figure 6.3: Using individual values to specify mappings

Apart from the changes in the layout of the mark design (when expanded, the mark's visual properties appear now to the left), this mapping process is pretty much the same as in the previous iVoLVER.

Figure 6.4 shows another sequence of mapping operations. Here, the process begins with the same mark used in the previous scenario (see step 1). In step 2 of this sketch, all the values of the `Name` attribute (available through the corresponding column header) are fed to the mark's `label` property. This connection leads to the creation of nine additional marks of the same `shape`, `height` and `color` that are distributed horizontally, to the right of the first mark (step 3).

The tool automatically adds two new visual properties—`X` and `Y`—to the original property set (step 4). These properties control the 2D position of the marks respect to the visualization's bounding box, which preserves the scale of the contained marks when resized (step 5). The visualization process of this sketch ends when the values of the `Salary` attribute are connected to the `height` visual property (step 6). This step modifies each mark's `height` with the respective values from the dataset (step 7).

The mark design described above is a variant of the Collective Proxy Objects family (discussed in section 5.6.2 of Chapter 5), in which an external widget allows to configure the visual appearance of a set of marks. However, in this design, the

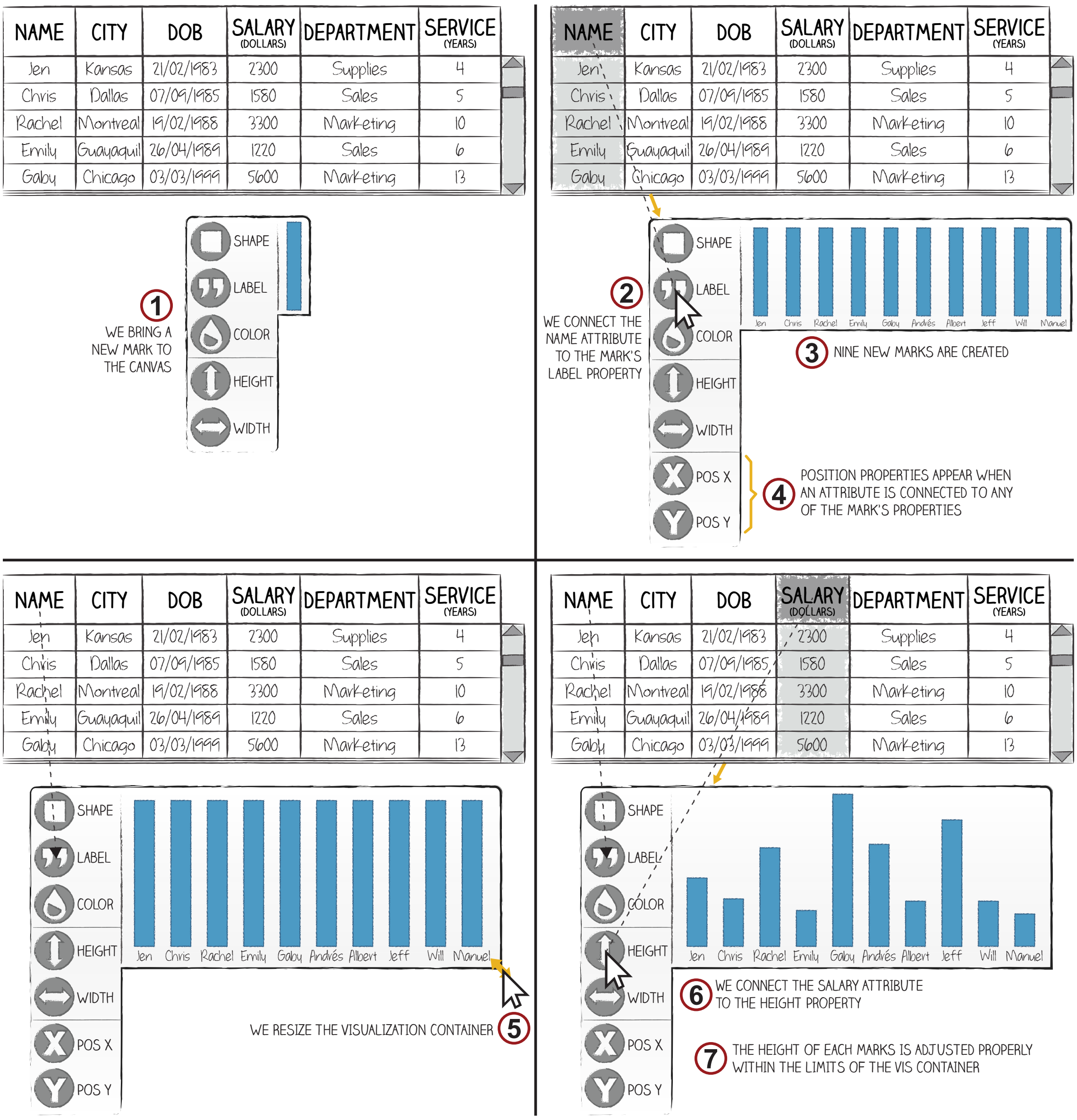


Figure 6.4: Using attributes to specify mappings

object used at the beginning of the process is a mark itself that becomes a proxy for others only when a *set* of values is connected to any of its visual properties. With this modification, iVoLVER supports now two different levels of atomicity for data manipulation (values and sets of these), which in turn leads to two levels of atomicity on the visual representation side (marks and visualizations).

To preserve flexibility and expressivity, the marks that compose a visualization can still be individually accessed through direct manipulation. However, in order to avoid loosing the semantic meaning of the visualization abstraction (as a set of marks), individual marks cannot be taken out of the boundaries of their visualization's container. If a single mark needs to be used out of its containing visualization, it should be cloned, but it cannot be removed.

By supporting additional higher levels of atomicity and abstraction, this version of iVoLVER tackles the scalability problem while still allowing the manipulation of atomic pieces of data and individual marks. This design also preserves the idea that a visual representation is also a configurator of its underlying data, avoiding the introduction of separate views (e.g., for configuration and rendering).

6.2.3.1 Combining Values and Sets

When mappings operations combine individual data values with sets of these (or entire attributes), the former are applied to all the marks of the visualization. Figure 6.5 shows an example of this. Ten marks are created in a single interaction

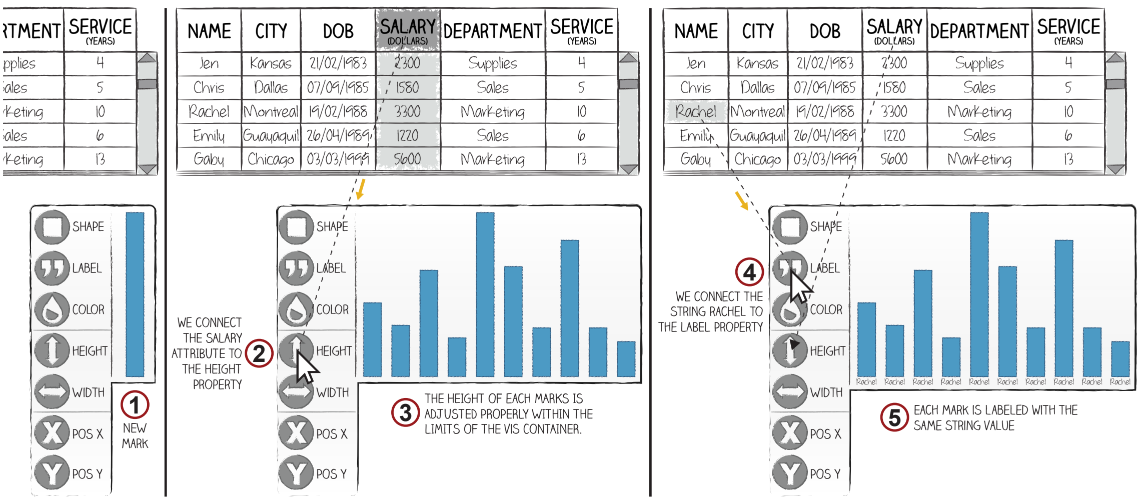


Figure 6.5: Using an individual value to modify a property in all the marks

step after connecting the `Salary` attribute to the initial mark's `height` property (steps 2 and 3). The string '`Rachel`' is then connected to the proxy's `label` property and, thus, set as the label of all the marks of the visualization (steps 4 and 5).

In scenarios where a set of values is connected to a mark containing mappings with single data values, the tool creates new marks and modifies their corresponding property with data from the dataset while keeping all the properties that use individual data values unchanged.

6.3 Data Transformations

The examples of Figures 6.3 and 6.4 involve connecting textual and quantitative values (or attributes) to properties that are suitable for these data types (`label` and `height`, respectively). Visualization tools that implement high levels of automation often allow designers to establish mappings between attributes and visual properties of "incompatible" types (e.g., mapping a quantitative attribute to a visual property that requires categories).

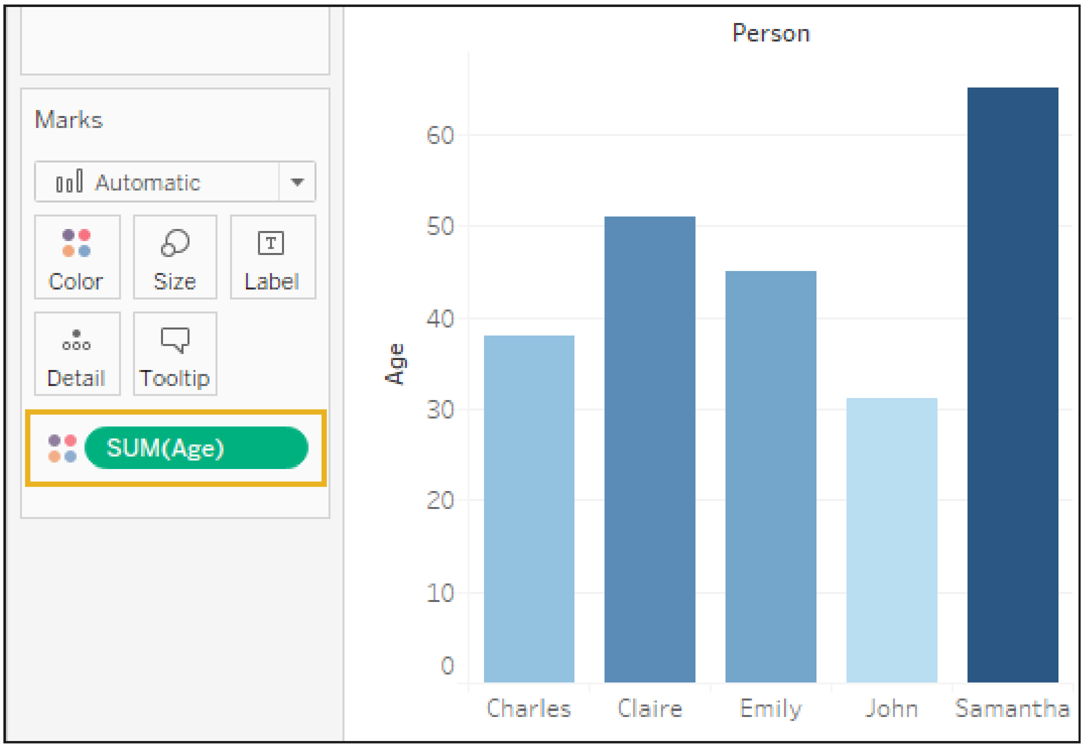


Figure 6.6: Color gradient created by Tableau from the `Age` numerical attribute

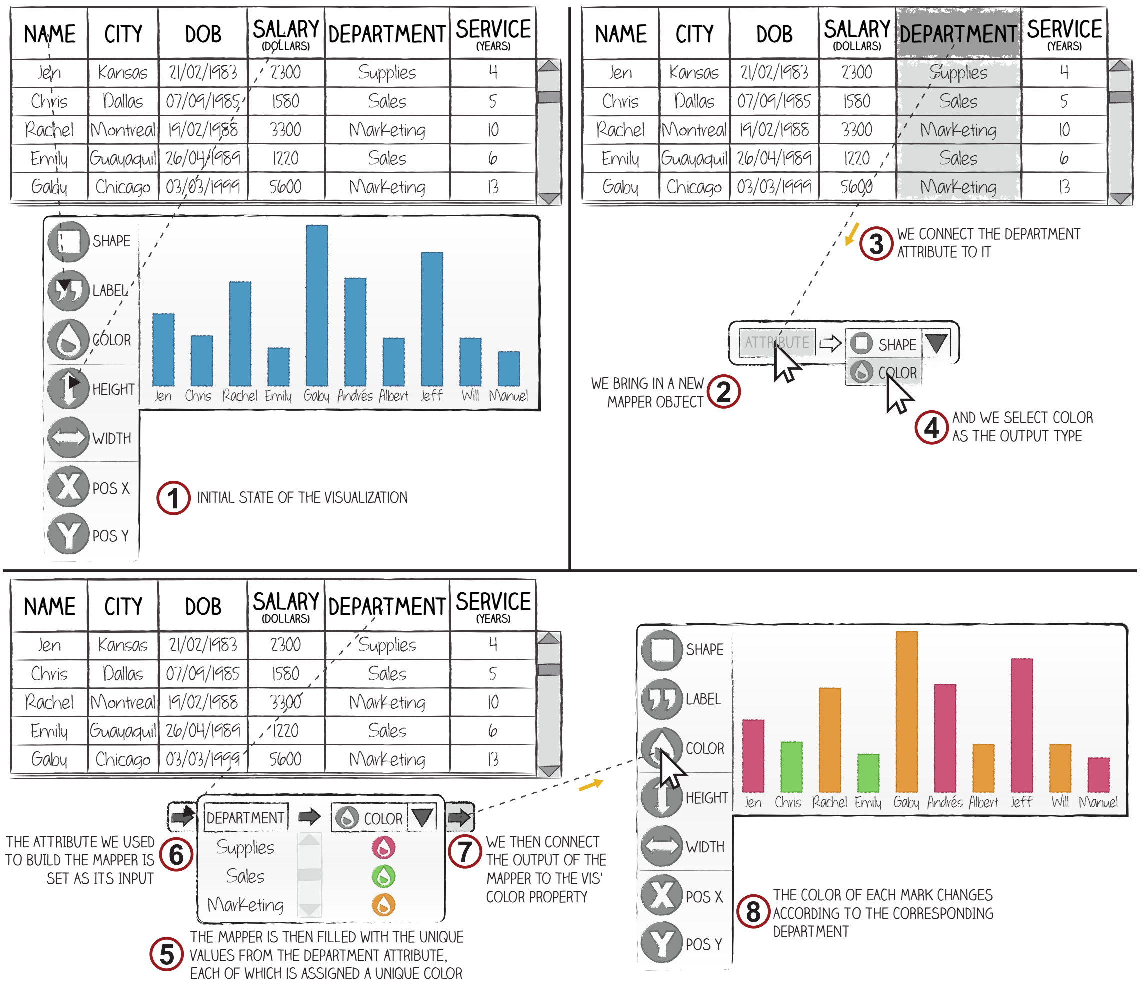


Figure 6.7: Use of a mapper widget to transform the values of a given attribute into colors

In Tableau, for example, it is possible to map continuous measures (i.e., numbers) to the `color` card. Figure 6.6 shows a Tableau visualization for the `People` dataset (see Table 4.3) that maps the `Age` quantitative attribute to the `Color` visual property. This steps automatically paints each mark of the visualization with a color from an automatically created gradient. The mapping itself is visually represented below the *Marks* card (see yellow rectangle in Figure 6.6).

Mappings that involve incompatible data attributes and visual properties are not possible in the current version of iVoLVER. The tool rejects, for example, connections between a string value and a mark's `color` property. This impacts the tool's flexibly and scalability, as it prevents users from mapping data attributes and visual properties that require an intermediate transformation step (e.g., converting numbers into colors, or strings into shapes). The revised iVoLVER presented in this chapter addresses this issue by using mapper widgets whose functionality resembles iVoLVER's original mappers (described in Section 3.4.7 of Chapter 3).

Figure 6.7 shows a visualization in which the `Name` and `Salary` attributes of the dataset define the `label` and `height` properties of ten rectangular marks (step 1). A mapper object is then brought to the canvas (step 2) and configured to transform the strings from the `Department` attribute (step 3) into color values (step 4). These steps populate the mapper with the unique elements of the `Department` attribute, each of which is associated to a different color (step 5). Additionally, all the values of the `Department` attribute are automatically fed to the mapper's input port and thus converted into colors (step 6). The result of this conversion is available through the mapper's output port that is finally connected to the `color` property of the visualization (step 7). This sequence of actions color each mark according to the corresponding `Department` (step 8).

The use of a mapper object solves the problem of connecting incompatible attributes and visual properties, enhancing the tool's flexibility. However, it also requires additional steps. The process, nevertheless, is transparent as it shows the intermediate data transformation step explicitly. To speed the process, the

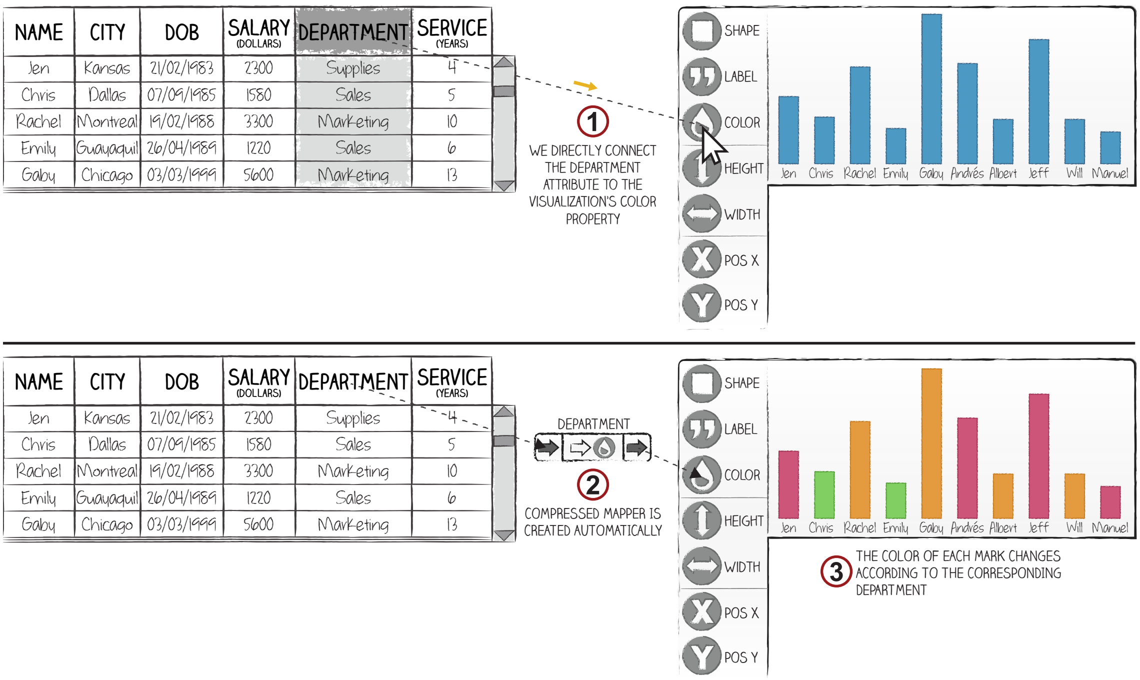
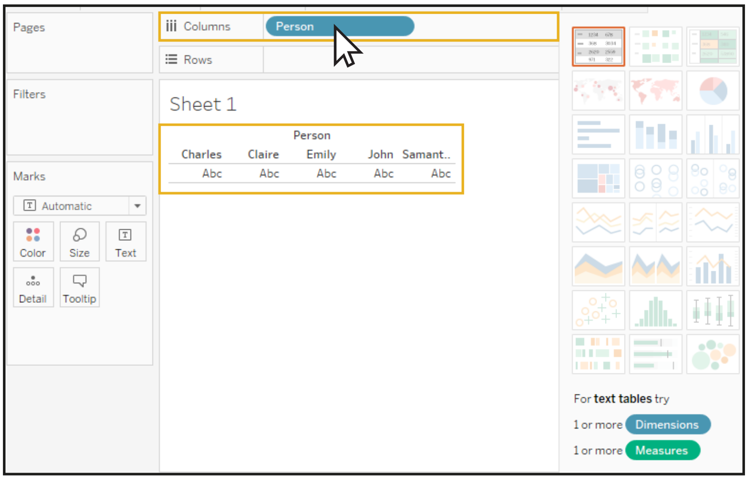
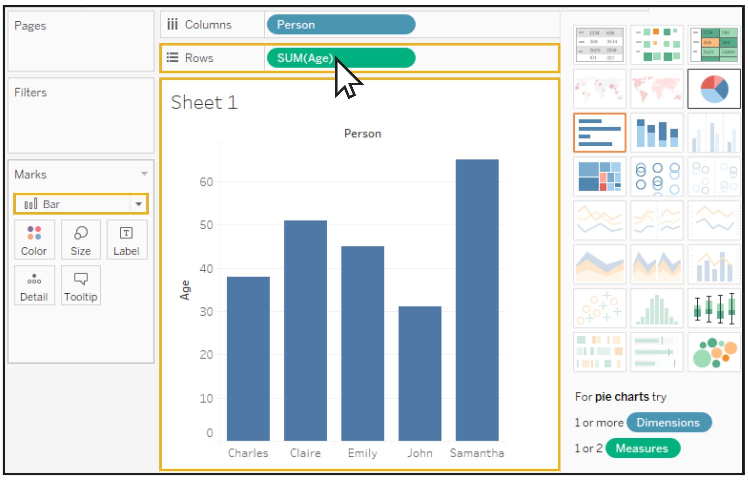


Figure 6.8: Automatic creation of a mapper to achieve a mapping operation that involves incompatible attribute and property types.

revised design of iVoLVER also allows for the automatic introduction of the mapper widget. Figure 6.8 shows the same initial visualization used in the previous example. In this sketch, mapping the `Department` attribute to the `color` property simply requires creating the corresponding connection (step 1). When the tool detects an unsuitable connection (that involves incompatible data and property types), it automatically creates a mapper that transforms the input values accordingly. The newly created mapper is added to the canvas and shown compressed (step 2). The tool also adds the corresponding intermediate connections to represent the final mapping.



(a) Dragging the `Person` attribute



(b) Dragging the `Age` attribute

Figure 6.9: Steps to build a barchart in Tableau. Yellow rectangles indicate the areas of the interface that change after the indicated interaction step.

6.4 Threats to Transparency

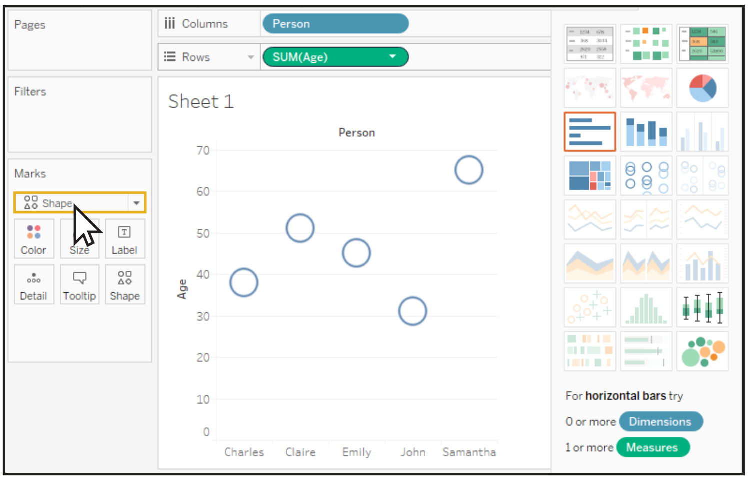
As discussed throughout this dissertation, one of the biggest drawbacks of excessive automation is that it harms the transparency of the visualization process because individual interaction steps might result in the application of several default behaviors, which in turn introduces in the visualization changes that could be difficult to identify, follow, and interpret. Ultimately, nontransparent changes have an impact on the users' perceived sense of agency and on their feelings of control and authorship regarding the visualization design process and its outcomes.

To illustrate this problem in Tableau, Figure 6.9 shows the steps required to build a bar chart in this tool. First, we drag the `Person` attribute to the `Column` shelf (Figure 6.9a), which produces the first version of the visualization: a table-like representation whose column headers are the unique elements of the `Person` attribute. We then drag the `Age` attribute to the `Rows` shelf (Figure 6.9b). This step introduces several changes in different parts of the interface. First, the table-like visualization is replaced by a bar-based representation designed by the tool. Second, the `shape` attribute in the `Marks` card is changed (from `Automatic` to `Bar`). Finally, some of the options of the "Show Me" pane are enabled.

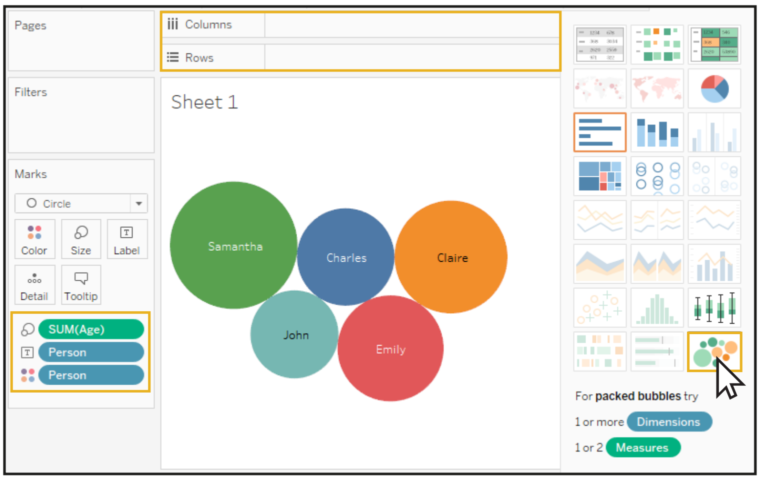
As the results of Study 2 suggest, transitions between individual interaction steps could be problematic, especially when they lead to fundamentally different outcomes (e.g., when the tool suddenly generates a bar chart from a table-based representation). These changes are often so drastic that not only harm the transparency of the visualization process but also affect the users' sense of agency (as the tool seems to be in charge of the process). Sudden, abrupt changes also make the visualization process look less incremental and, in consequence, less constructive.

As demonstrated by the example of above, the changes that take place when creating a new visualization in Tableau can be sometimes hard to track. However, editing an existing design could be even less transparent. Figure 6.10 shows two visualizations that result after two independent modifications of the bar chart of Figure 6.9b. Both visualizations can be achieved in Tableau Desktop with a single interaction step. In Figure 6.10a we modify the marks' `shape` property via the *Marks* card. This step replaces the previous rectangular marks with non-filled circular ones whose `vertical positions` encode the `age` of each person. In this example, modifying the `shape` property also changes the marks' `color` and introduces a specific position mapping. On the other hand, the visualization of Figure 6.10b shows the result of selecting the "packed bubbles" design from Tableau's "Show Me" pane. This second outcome also involves circular marks, but these are now colored by person and arranged to form a packed structure (although this positioning is not data-driven). The tool also removes the attributes previously dragged onto the "Columns and Rows" shelves.

As in Tableau, transparency could also be affected in the revised iVoLVER because of the higher level of abstraction it now supports. Given that attribute-level operations are now possible, some interaction steps have consequences on all



(a) Changing the shape visual property



(b) Selecting a bubble chart design.

Figure 6.10: Results of two independent changes made to the bar chart visualization shown in Figure 6.9b. Yellow highlights indicate the areas of the interface that change after the indicated interaction step.

the marks of the visualization. Additionally, the modification of specific visual properties will necessarily require the application of default settings.

Consider, for example, the sequence of Figure 6.11 that shows how a bar chart is transformed into a bubble chart-like visualization. To promote a transparent visualization process, the revised iVoLVER requires the designer to execute a few more steps to achieve results that in Tableau would only take a single click. The tool still requires users to drive the process, but also provides the needed built-in automated repetition. The designer first needs to modify the shape attribute of the visualization (step 1) by selecting the new shape from a list of built-in options (step 2). The tool then converts the existing rectangles into circular marks in a process that preserves all the properties that are not related to `shape` (e.g., `label`, `color`—step 3). Finally, the designer manually rearranges the marks to achieve the desired layout (step 4).

The process depicted in Figure 6.11 requires the tool to apply at least two default settings when converting a single rectangular mark into a circular one. First, it

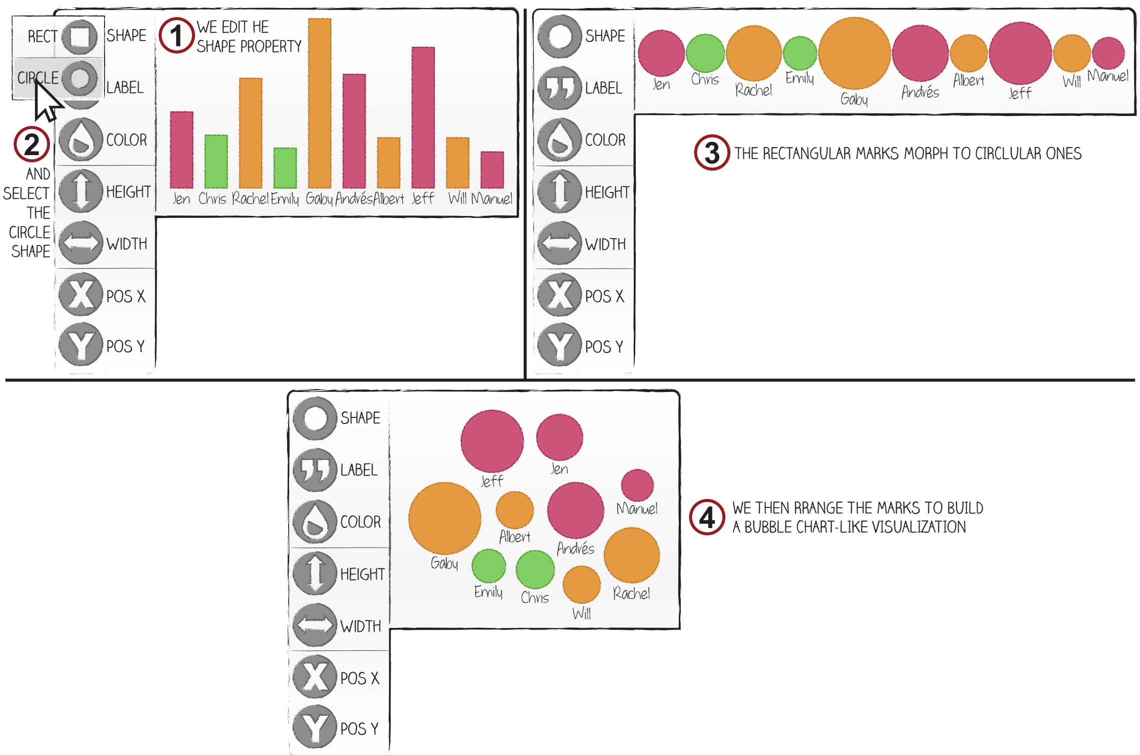


Figure 6.11: Converting a bar chart into a bubble chart visualization in the revised design of iVoLVER

preserves the area of the original rectangles and uses it to decide the size of each new circle. A rectangle of height h and width w results in a circle of radius equal to $\sqrt{(w * h)/\pi}$. The tool also decides the position of the new marks. In the depicted sketch, all the circles' center points are aligned and this positioning also determines the location of their labels (see step 3 in Figure 6.11).

The process described above is arguably more transparent than its Tableau counterpart, but it still imposes some threats to transparency. First, replacing existing marks with new ones of a different shape could be an abrupt change that non-experts might have problems following. Second, the application of default settings when specific visual properties are changed might not be quite evident. Along these lines, preserving the `area` might not be the best option for the rectangle-to-circle transition. Each default setting will have specific implications on the perceived transparency and agency of the design process.

6.5 Discussion and Open Issues

The revised iVoLVER enhances the scalability of the constructive visualization process promoted by the tool's original version. This is achieved by supporting interactions with the data and the visual representation at lower levels of atomicity. The revised design supports operations at the attribute-level (not only with individual values) and interaction with the visual construction as a whole (not only with the marks that compose it).

These features, however, impose the need of automated repetition and automated choices (i.e., application of default settings). The presented design applies these strategies while aiming to achieve a user-driven design process that is as transparent as possible. However, as discussed in Chapter 4, the trade-offs are quite tricky and intertwined. Any of the applied strategies increases scalability but could also place too much agency on the tool side and affect its constructive nature.

6.5.1 Alternative Solutions

As stated at the beginning of the chapter, the design here presented is one of many that could achieve scalability in iVoLVER. In particular, this solution attempts to preserve most of iVoLVER's original ideas without altering the interaction with the tool at a fundamental level. For example, regardless of the atomicity of the data involved, mappings are still achieved with drag-and-drop gestures that begin on the dataset and end in a mark's visual property. This allows designers to transition seamlessly between manipulation of individual values and sets of these, without having to execute additional steps (such as special gestures or activating modes).

However, other solutions could achieve similar results by applying the strategies used here in different ways. For example, the automated repetition that this design implements takes place incrementally, as designers carry out new mapping operations. Although a macro recorder solution (based on a by-example strategy) would also introduce automated iteration, it would execute it in bulk, after a series of several mapping operations. In the revised iVoLVER design presented in this chapter, I decided in favor of incremental automated iteration to further preserve the constructive nature of the tool.

6.5.2 Mechanisms to Preserve Transparency

Additional mechanisms to preserve transparency could be introduced in the steps in which the tool applies automated choices. For example, modifications of the shape property could be animated to reduce the potential impact of sudden changes in the visualization. Heer and Robertson investigated the design of animated transitions between a specific set of visualizations and found that, if designed appropriately, "*animated transitions can improve graphical perception of changes between statistical data graphics*" [82, p. 1247]. More general research has also shown that animation can keep people oriented [171, 209] and that it can increase levels of engagement [209]. Considering these observations, I believe that using animations at the mark level (rather than at the visualization one) could help to preserve the transparency of the process. The alternatives include morphing shapes, smoothly transitioning colors, and tweening changes in size-related mark properties. These ideas, however, must be investigated in experimental settings. After all, animation introduces a level of complexity that static representations do not have.

6.5.3 Aggregating Data

In InfoVis, aggregation is a common way to represent large amounts of data by replacing individual data points with representation of subsets. This type of mechanisms sound promising to further increase the scalability of digital constructive visualization tools such as iVoLVER. However, aggregating implies, by definition, encapsulating (i.e., hiding) which is in direct opposition to the visual explicitness principle of iVoLVER and to the idea of enabling high levels of atomicity. I suspect that it would be feasible to implement widgets that make data aggregation processes more visually explicit and, thus, more transparent (like the mapper presented in section 6.3). Further design iterations could explore this type of solutions and the advantages that they have (if any) to increase the tool's overall scalability.

6.5.4 Visual Representation of Large Canvas Objects

The visual explicitness principle that drove the design of iVoLVER aimed at avoiding hidden elements both in its UI and in its visual programs. This principle also enables making some components of the visualization design process visually explicit. For example, iVoLVER represents the visual mappings themselves through connections between the data tables and the proxy objects' visual properties. Other mapping-related operations, such as the representation of data transformation through mappers (see Section 6.3), also make specific steps of the visualization design process visually explicit on the tool's canvas.

The visual explicitness principle has, however, serious implications on how the screen real estate is used in the tool, which is a critical aspect of any visual programming environment. The revised design presented in this chapter addresses the issue of graphically representing large datasets on the screen by introducing scrolling and resizing (see Section 6.2.1). These strategies allow to show data on demand, as explicitly requested by the visualization designer. However, they might also require a significant number of interactions with the visual representation of the dataset, especially when trying to access records near the end of the table.

In general, scrolling and resizing could be applied when large collections of objects need to be visually represented on iVoLVER's canvas. These strategies could be incorporated, for example, in the design of the mapper objects used to transform data (see Section 6.3) or in the visualization containers that proxy objects create.

New design iterations could explore other ways to cope with the issue of representing large canvas objects. It is also important to investigate whether making all the elements of a collection (e.g., all the rows of a data table) visible is essential or even beneficial in digital constructive visualization tools. It could be argued, for example, that showing every single data point of a large dataset might be unnecessary in the revised design, as accessing the dataset's attributes (through the column headers) might be enough to carry out the visualization process.

6.5.5 Flexibility and Scalability

In the revised design of iVoLVER, the marks that compose a visualization are still accessible individually. In the sketch of Figure 6.11, for example, marks are manually manipulated (i.e., dragged) to achieve a specific spatial arrangement. As found in Study 2, this direct manipulation leads to a more flexible process and enables the expressivity principle of ConstructiveVis. This strategy, however, does not scalable well to larger sets of of marks. New interactions techniques are still needed to support scalable manipulations of marks. It is also important to acknowledge that this chapter does not go all the way to make the design fully scalable. As discussed in the previous subsections, there are some scalability issues that remain unsolved.

6.5.6 Need of Empirical Validation

It is still unclear whether this revised version of iVoLVER would exhibit the same benefits (and limitations) than its original version. In some ways, the solution this chapter presents incorporates several Tableau-like functionalities but with increased transparency and less arbitrary—or enforced—involvement of the tool. Additional investigations are still needed to assess this design from an experimental perspective.

6.6 Conclusion

This chapter presented a revised version of iVoLVER that integrates the lessons learned from the studies conducted in the preceding chapters to make the tool more scalable. The presented design incorporates automated repetition, includes a higher level of abstraction, and implements automated choices for specific parts of the visualization process. The design applies these strategies as an initial step to preserve a user-driven and transparent visualization process that is still constructive but more automated than the one of iVoLVER's original version. This chapter also discusses the proposed design and shed lights on how some open issues could be addressed in an actual implementation.

Chapter Seven

7 Discussion

The aim of argument, or of discussion, should not be victory but progress.

— Joseph Joubert

This chapter presents a summary of the findings of this research and interprets the overall results and implications. It also summarizes the limitations of this work and discusses the questions that remain open. The chapter ends with a summary of guidelines for visualization tools designers and a critical reflection.

7.1 Summary of Findings

The preceding chapters presented and discussed several findings around the design, implementation and use of iVoLVER, a digital tool that promotes a bottom-up, constructive approach to visualization authoring. This dissertation described three studies that investigated the use of this tool from different perspectives. Study 1 (Chapter 3) looked at the usability of an early version of iVoLVER. The outcomes of this study motivated several modifications within the iterative approach I took to design and implement the tool.

Study 2 (Chapter 4) compared how two tools that implement opposite approaches to visualization construction influence non-experts' visualization authoring process. I studied iVoLVER and Tableau Desktop as representatives of the ends of a continuum spanned between bottom-up and top-down approaches to visualization authoring. The study showed that Tableau Desktop enabled the speedy exploration of visualization solutions with minimum input from the user, but it did not promote a strong understanding and critical reflection on the visualization process and outcomes. On the other hand, as a constructive tool, iVoLVER required initial thinking about the intended visualizations and a hands-on, sometimes tedious procedure of mapping data to visuals. This resulted, however, in a transparent, flexible visualization authoring process that facilitated creative and critical thinking as well as a stronger feeling of control and authorship.

I further explored the design space of constructive visualization tools. To this end, Study 3 (Chapter 5) involved a mixed methodology that combined the outcomes of an iterative in-house design exercise with the ideas from a group of people who took part in a series of design workshops. This study investigated how to design tools that support a user-driven, transparent design process while enabling efficiency and automation. The outcomes of the study ranged from example-based approaches that directly expand ConstructiveVis to designs in which the tool infers visualization solutions on behalf of the designer (e.g., based on data attributes). Further analysis on this data led to a characterization of the design space of visualization tools in terms of agency and atomicity.

Finally, the outcomes of Studies 2 and 3 drove the revision of iVoLVER's original design, presented in Chapter 6. This new design allows access to data and visual representations at an additional lower level of atomicity (supporting operations with attributes, affecting the entire visualization with individual interaction steps). The new design increases the tool's level of abstraction, requires the implementation of automated repetition, and introduces automated choices for the steps in which the tool needs to apply default settings.

7.2 Overall Discussion

This dissertation sheds light on the implications that constructive principles have when integrated in digital visualization tools and contributes insights on the challenges that arise in the design and implementation of such tools. I supported my research in the area of digital ConstructiveVis with a visual programming environment. Although it would be unfeasible to establish a one-to-one correspondence between the design of iVoLVER and the features of the tangible tokens originally used by Huron [93, 95, 97], taking a visual programming approach supported an important aspect of the constructive process: the direct manipulation of atomic pieces of data and individual visual marks. This, in turn, allowed me to study iVoLVER in settings comparable to the ones used by Huron in his initial investigations of constructive visualization processes (e.g., [97, 223]).

The findings of Study 2 show that iVoLVER, as a digital constructive tool, promotes a mindful visualization process that encourages users to reflect on their design approach. The outcomes of comparing iVoLVER and Tableau Desktop are largely explained by the theoretical foundations of ConstructiveVis. In Papert's vision, for example, knowledge is a personal experience to be constructed, not merely a commodity to be transmitted [3]. Study 2 supports that it is precisely the hands-on nature of the process iVoLVER promotes that leads to the observed benefits, as it requires users to actively engage in the design of the visualization. The findings show, however, that this can also be perceived as tedious and unnecessary, as people often expect the tool to do more on their behalf. I see this as a "desirable difficulty" in the sense defined by Bjork: as a task that requires a considerable but desirable amount of effort [18]. Research has shown that introducing desirable difficulties in educational material can make learning more effective [17, 19]. These principles have also been investigated in InfoVis. Hullman et al., for example, showed that introducing visual difficulties in visualizations can, to a certain extent, benefit comprehension and recall [91].

Based on Bjork's theory, the desirable difficulties iVoLVER's imposes in the visualization process can indeed be helpful. However, to be desirable, a task must also be accomplishable. In this regard, iVoLVER's need of repetitions of gestures and actions represents a critical issue that could make the whole enterprise of digital ConstructiveVis unpractical: the construction process does not scale well to larger datasets. Making iVoLVER (and possibly any digital constructive visualization tool) scalable, without sacrificing the observed benefits, could be challenging. iVoLVER's design principles (e.g., atomicity, visual explicitness) unavoidably lead to trade-offs that, if unbalanced properly, can sweep away the constructive nature of the tool entirely.

The scalability problem of iVoLVER is mainly a consequence of the individual manipulation of "atoms" of data and visual representations that the tool requires. Although problematic, this is a fundamental characteristic of ConstructiveVis as it allows for an incremental (i.e., constructive) process in which the visualization emerges a consequence of the user's actions. There is also some evidence to support that iVoLVER's desirable difficulties also explain the increased feeling of control and authorship that participants experienced when working with it.

The results of the visualization exercise conducted as part of Study 3 confirm the previous observations. As in Study 2, participants perceived as disadvantageous that Tableau takes the design lead right after they identified—and dragged out—the data attributes involved in a given visualization task. iVoLVER's scalability problem was equally evident, popping out after just a few interactions. One participant actually identified the problem during the tool training phase of the workshop asking "*How can I make it to repeat the same mapping for all the data points?*".

Study 3 leveraged participants' contrasting experiences with iVoLVER and Tableau to motivate ideas to make iVoLVER able to support visualization of larger datasets, without loosing the benefits of its constructive principles. This study focused on mapping the design space of digital constructive visualization tools and uncovered three families of potential solutions: visualization by example, collective proxy objects, and automated principled design. The ideas that compose these families span across the bottom-up top-down continuum of approaches to visualization authoring described in Chapter 4 of this dissertation. Each family is driven by a different underlying strategy: automated repetition, abstraction, and automated choices, respectively.

As mentioned earlier, given the empirical evidence gathered throughout this research, I suspect that tools located in Q_1 of the design space presented in Chapter 5 could address the scalability problem of ConstructiveVis while preserving most of its benefits. In fact, very recent research within the InfoVis community has started to explore visualization authoring tools based on a by-example strategy. In VisExemplar [176], Saket et al. support exploration of data by allowing users to provide visual demonstrations of incremental changes to the visual representation. From these manipulations the tool constantly recommends potential data transformations that users can apply at any time. In line with iVoLVER's design principles, VisExemplar supports the direct manipulation of visuals and implements automated iteration. Although in this type of design the tool still contributes to construct the visualization, its role is providing guidance to the user in response to their actions. Unlike Tableau Desktop, this design does not enforce its suggestions or execute them without the users' request. In this regard, it is similar to the in-house "Smart Assistant" presented in section 5.3.2 of Chapter 5.

The revised design of iVoLVER presented in Chapter 6 achieves a more scalable visualization process that is still transparent and user-driven. This, however, required increasing the tool's level of abstraction (see Section 6.2.3), implementing automated iteration and applying automated choices. Although the introduction of these strategies can potentially decrease the level of atomicity and shift the agency balance towards the tool side, the design preserves most of iVoLVER's original principles. In fact, this version still supports interaction at the most atomic level (individual data values and marks), allowing a visualization process with the same characteristics of its original version. In addition, it supports visualization authoring at less atomic levels, allowing users to perform operations at an attribute and visualization levels. Empirical evaluation of this design is still needed to get insights on its impact on non-experts' visualization process.

One of the main outcomes of Study 3 is the design space defined by agency and atomicity, dimensions that specify, respectively, *who* drives the visualization process (i.e., the designer or the tool) and with *what* type of elements (i.e., atoms or higher-level abstractions). The aspects of atomicity and agency are key to the paradigm of Constructive Visualization. As the findings of Study 2 suggest, atomicity is essential to achieve a constructive process, as it allows the execution of incremental steps. On the other hand, given that constructionism focuses on personal experience as the gateway to understanding and reflection, careful analysis of agency is fundamental to achieve visualization processes centered on the designer (not on the tool). After all, Papert's theory is concerned with contexts "*where the learner is consciously engaged in constructing a public entity.*" [152, p. 1].

This research provides insights on how the concepts of agency and atomicity conceptualize the problem of scale in any visualization tool (not only constructive ones). These dimensions provide a lens to look at existing approaches and strategies and, on a more practical level, bring ConstructiveVis into a more holistic perspective providing common ground to discuss it in the wider context of the current visualization tools landscape. The characterization of the existing landscape of visualization tools in terms of agency and atomicity is complementary to others, such as the top-down vs. bottom-up continuum of visualization authoring approaches discussed in Chapter 4 or Tory and Möller's taxonomy [206].

The empirical evidence gathered through this research shows that the lack of visualization expertise in analytical scenarios does not have to be tackled, necessarily, with black box-like solutions that do (almost) everything on behalf of the designer. There is a tendency within the software development and HCI communities to build solutions that are fast, efficient, and prevent users from making mistakes—so much so that we often measure the quality of a software solution in terms of times and errors, both of which we try to minimize. However, characterizing the effectiveness of visualization tools from the user's perspective goes beyond usability metrics [13, 14]. The activities that take place in analytical settings are quite complex and, thus, require other types of assessment.

These challenges have been discussed within the InfoVis community for over a decade [158] and some efforts have explored ways to focus the evaluation of visualization tools on understanding [161, 182] rather than just usability. Saraiya et al. [161], for example, investigated insight generation in the long-term usage of a visualization tool.

ConstructiveVis is an attractive alternative to the tools that promote a no-need-to-think philosophy, not only because it supports the creation of effective visualizations but also because it promotes critical reflection on the steps involved in the process. In turn, this not only democratizes visualization technologies but the design process itself. An even more exciting alternative is a scalable visualization tool in which users can transition between the two ends of the bottom-up vs. top-down continuum of approaches to visualization authoring according to their level of expertise. The revised design of iVoLVER partially achieves this, as it allows the manipulations of both atoms (individual values and marks) and higher abstractions (sets of values and visualizations). In a more general sense, the implementation of such tools should allow to adjust the required level of atomicity and agency according to the users' needs or expertise (or both). In any case, digital ConstructiveVis should not be seen as a one-size-fits-all approach. Its application and benefits should always be contextualized based on the users' expertise and the pursued goals.

7.3 Limitations

The findings this dissertation reports are recognizably limited by the research methodology followed in the conducted studies. Although I chose study methodologies that fit the purpose, no methodology is perfect and we need to be aware of the limits of the knowledge obtained by this research.

I used Tableau and iVoLVER to investigate the visualization process of non-experts at the extremes of the bottom-up vs. top-down continuum of approaches to visualization authoring. Although both tools are comparable in the interaction steps they require to build visualizations (e.g., simple mouse operations, drag-and-drop) and the concepts they use (e.g., marks and visual properties), each has a specific workflow. The design of this study considered Tableau and iVoLVER as specific lenses to investigate the ends of a continuum of visualization authoring approaches. Evidently, the study could have included other tools with different interfaces and workflows. However, as discussed in Chapter 4, the observations and findings of this study reflect on the underlying approach that each tool follows rather than on its specific implementation. Most of the reported observations are about high-level concepts that any visualization tool would exhibit and that are not interface-specific. For example, participants identified the importance of the visibility and flexibility of the visualization design process regardless of the particular implementation of each tool. Nevertheless, further evaluation on how other concrete tools also located at the ends of the bottom-up vs. top-down continuum are needed to confirm the validity and generalizability of the results here presented. It is also important to highlight that the comments gathered through interviews are about participants' self-reported experiences. Thus, they are subjective and might be background-dependent.

Similarly, the ideas gathered through the design workshops to solve the problem of scalability in digital constructive tools (Chapter 5) should be seen as a catalogue of solutions limited in several aspects. First, it is clear that participant's experience with Tableau and iVoLVER influenced their motivations for the sketching exercise. In their solutions, they often focused on reducing the level of effort that constructing visualizations in iVoLVER required. Furthermore many of their designs were inspired in tools they had used in the past (e.g, MS Excel) or even in Tableau Desktop itself, because of its familiar look-and-feel. There is also a clear bias in the participants' designs towards the use of conventional WIMP-like design which, by definition, is in opposition to some of iVoLVER's design principles and to its underlying visual programming approach. Finally, although I explicitly set out to include people with different backgrounds and levels of expertise, the study's sampling is small and not fully representative of all potential audiences. Additional studies are needed to investigate approaches of other populations (e.g., professionals).

7.4 Open Questions and Directions for Future Research

Several questions remain unanswered regarding whether the observations here reported hold for users, scenarios, settings, datasets, or tool implementations that are different from the ones used in this research. The following subsections reflect on some of these questions and provide directions for future investigations.

7.4.1 Benefits for Expert Users

ConstructiveVis was born as a potential solution to the problems that non-experts find when working with more conventional visualization tools. The findings reported in this dissertation confirm Huron's early observations on ConstructiveVis with physical tokens. However, both Huron's experiments and the studies conducted in this research involved participants with no expertise or formal training on InfoVis concepts. Because of this, the findings reported in this dissertation cannot be generalized to other populations or analytical scenarios in which visualization expertise can be assumed.

This dissertation discusses ConstructiveVis as a way to scaffold and clarify visualization principles while engaging non-experts with visualization. However, this research does not touch on the following questions: *Would more experienced users perceive a constructive approach as beneficial as non-experts did? What kind of limitations would they find when working with constructive tools? How could digital constructive visualization tools be adapted to support authoring process of more expert users?*

7.4.2 ConstructiveVis Without Visual Programming

Using visual programming to implement iVoLVER as a digital constructive tool supported a visualization process in which users could establish mappings by directly manipulating both the data and the visual marks. Aiming at integrating data representation and configuration within a single space, in my research, this construction process took place within an open canvas. The elements of this canvas are not only representations of some underlying data, but also the gateway to configure how such data is visually represented.

It is safe to assume that other visualization tools could achieve constructive features, for example, by implementing interfaces based on the WIMP paradigm. *Could these tools bring the same (or even additional) benefits that this dissertation identified into the visualization process of non-experts?* Along the same lines, *which design principles should guide the development of such tools? How would their visualization process compare to the one of iVoLVER?*

7.4.3 Benefits Beyond Critical Reflection

The insights gathered in this research show that digital constructive visualization tools have the potential to promote reflection on design principles while engaging non-experts with visualization. The main aim of exploring data with the support of visual representations, however, is the generation of insights, learning, and reflection on the manipulated data itself. That is, data exploration processes should help people to identify patterns and raise questions on the analyzed data.

Along these lines, Huron's investigations with tangible tokens, showed that people "*spontaneously manipulated their visualizations to support their arguments*" [97, p. 2108] about the datasets. This suggests that tangible ConstructiveVis could indeed contribute to the generation of richer insights on the data. Future research is needed to investigate if these observations hold for digital constructive visualization tools. Assuming that scalability is not a problem, *could a constructive tool lead to the generation of richer or more insights on the visualized data?*

7.4.4 Digital ConstructiveVis with Non-Tabular Datasets

All the investigations conducted as part of my research involved the use of tabular data. I decided in favor of this type of dataset because they are widespread in most areas of science and education. Hence, even non-experts would be familiar with the way information is organized within a table. However, visual data analysis also takes place with other types of datasets, such as networks [55, 73, 191] or textual information [101, 186, 188].

Given that ConstructiveVis enforces users to get involved in the process of mapping data attributes and visual properties, it would be interesting to explore whether this visualization paradigm can also support the construction of scientific visualizations, in which spatial encodings are mostly driven by the nature of the data itself. Tangible representations of non-tabular datasets have been explored before, for example, to model proteins using plasticine [109]. However, the question in the digital constructive realm remains open: *how could digital constructive tools support visualization of datasets beyond tabular data?*

7.4.5 Visualization in Opportunistic Settings

Besides contributing to materialize ConstructiveVis into a digital solution, the design of iVoLVER also sought to contribute to democratize visualization technologies in casual and opportunistic scenarios where the data does not exist in structured database-like documents. By including in iVoLVER several types of data extraction capabilities, I tried to make visualization available in a wider set of scenarios. This included support for informal, casual and opportunistic data analyses that take place without anticipation, away from the desk, and that might involve the use of digital artifacts encoding unstructured data. Data extraction functionalities, however, were not evaluated when studying the constructive features of the tool. When Tableau was used as a counterpart of iVoLVER, it was necessary to reduce the scope of the problem exclusively to visualization tasks in order to ensure the ecological validity of the conducted studies.

Although I believe there is value and potential in supporting data extraction in visualization tools, as also evidenced by a large body of recent research in this area (e.g., [46, 75, 76, 106, 105, 159, 224]), this sort of investigations deserve their own space. Future research endeavors in this area could investigate: *What are the effects of supporting ConstructiveVis in a wider range of scenarios (e.g., opportunistic ones)? Would tools with data extraction capabilities encourage more people to engage with visualization technologies for their day-to-day activities?*

7.5 Guidelines for Tool Designers

The studies described in Chapters 4 and 5 report specific insights that can guide the design of new and better visualization tools. This section provides a brief summary of the most relevant guidelines for visualization tools designers according to the findings of these studies:

- Designers should seek a healthy balance between the user's and the tool's decisions.
- Although a speedy visualization construction process is generally desirable, speed may also be harmful as it can reduce transparency.
- Black box-like designs should be avoided; they can lead to the notion of "no need to think".
- Tools with many default behaviors can achieve ease of construction, but often at the cost of flexibility and transparency.
- Default behaviors can also reduce the users' sense of agency (including ownership and authorship of the process' outcomes).

- Abrupt changes in spatially separated sections of the interface could make users to feel lost after carrying out a particular action.
- Designers should aim at tools that overcome ConstructiveVis' inherent scalability problem. Non-scalable solutions (i.e., those that require excessive amounts of time, effort, and repetitive interaction steps) will harm people' willingness to explore alternative visualization designs.
- Three strategies may be applied to achieve scalable digital constructive visualization tools: automated iteration, abstraction, and automated design.
- Any of the strategies that achieve scalability mentioned above will have an impact on the agency of the process a tool supports. Designers must consider their implications when choosing which one to implement.

7.6 Critical Reflection

In the past few decades, InfoVis and visual literacy have started to play an increasingly important role in people's lives. Regardless of our background or what we do for a living, we are constantly surrounded and exposed to data representations. From activities like reading the newspaper, to browsing social networks, to walking inside a venue with posters, we are exposed to objects that present us data in one way or another. Examples include visualizations depicted in the print media, infographics works published in blogs, tabular data contained in web pages, and photographs of statistical charts.

Recent developments in sensing technologies have actually shifted the role of non-experts from being simply passive observers of visualizations that other people make to a more active role in which they not only generate data but also interact with it to understand themselves and gain several types of insights [88, 168]. Similar technological advances have even motivated research on how also children can benefit from visualization [6, 134].

The advances in the democratization of visualization technologies call for a redesign of how visualization authoring is currently supported. This dissertation contributes with work in this direction. By using constructive principles originally explored in the context of InfoVis by Huron and colleagues [93, 95, 97, 223], I have studied how to design alternatives to the predominant visualization authoring strategy that heavily relies on automation and pre-defined built-in solutions (e.g, templates).

This type of tool design can impose fundamental problems for non-experts, as often the tool takes the design lead. Previous research from data design has shown that active involvement [148], and repetition [15] can foster creativity and critical thinking. Designing tools that promote critical reflection on visualization principles instead of simply orchestrate most of the process is one of the goals that motivated this research. As Nicholas Carr wrote, "*The real danger we face from computer automation is dependency. Our inclination to assume that computers provide a sufficient substitute for our own intelligence has made us all too eager to hand important work over to software and accept a subservient role for ourselves*" [36]. Although the quote refers to the broader discussion about general replacement of human skills, it is also relevant for visualization design. After all, the tools that we design aim at helping people in a process that, by definition, augments cognition through the use of visual representations [35]. Hence, there is value in designing tools that encourage users to critically think on their design decisions. Ultimately, computational tools for visualization tasks should support users' cognitive process, not replace it.

Although this work does not provide a comprehensive analysis of all potential design considerations around digital constructive tools, it contributes insights that further our understanding on this area. My investigations on how digital tools that incorporate constructive principles shape people's visualization approach can inform the design of new tools both from an InfoVis and a HCI perspective.

Throughout my research, I have identified benefits, limitations and challenges that arise when working in this design space. Some of the findings of this dissertation have been published in conference papers and are starting to direct attention of members from the InfoVis community to the issues here addressed. I hope that the contributions of this dissertation help visualization researchers and tool designers to push forward the democratization of visualization technologies and to design better visualization tools.

Chapter Eight

Conclusion

A graphic is not only a drawing; it is a responsibility, sometimes a weighty one, in decision-making. A graphic is not 'drawn' once and for all; it is 'constructed' and reconstructed until it reveals all the relationships constituted by the interplay of the data. The best graphic operations are those carried out by the decision-maker himself. A graphic is never and end in itself; it is a moment in the process of decision-making. To construct a useful graphic, we must know what has come before and what is going to follow.

— Jacques Bertin [11, p. 16]

This dissertation provides an analysis of the design and study of digital visualization tools that incorporate constructive principles. The findings of this research also further our understanding on the role of ConstructiveVis within the broad design space of visualization authoring tools. On a practical level, the insights here described can guide the design of better visualization tools that not only allow people to create effective visualizations but also promote critical reflection and support a better sense of authorship of the process and ownership of its outcomes.

8.1 Answers to the Research Questions

To further our understanding of how ConstructiveVis principles could be integrated in digital visualization tools, this research set out to address several research questions (listed in Section 1.2 of the Introduction chapter). In this section, I briefly summarize the answer to each of these questions as discussed throughout this dissertation.

- ***Q1: Is it possible to design and implement digital constructive visualization tools?***

  Yes, it is. The implementation of iVoLVER, described in Chapter 3, shows how a digital visualization tool can incorporate constructive principles.

  Q1.A: What kind of principles could drive the design of such tools?

  If supported by visual programming, a digital constructive tool should: (1) seek atomicity by making small pieces of data and visual marks accessible through direct manipulation, (2) make the interface elements and the process as visually explicit as possible, and (3) avoid menus and hidden operations. See section 3.2 of this dissertation for more details.

  Q1.B: What challenges arise in the design and development processes?

  There is a complex set of intertwined trade-offs that emerges when building a constructive visualization tool (e.g., user- vs. tool-driven design, ease of construction vs. flexibility, speed vs. transparency). Balancing these trade-offs is critically important as, if done improperly, they can remove the constructive nature of the tool. See sections 3.8 and 4.7 for more details.

- ***Q2: How do digital constructive tools compare to others implementing a more conventional visualization authoring strategy?***

  As explained in Chapter 4, opposite approaches to visualization construction shape people's visualization process in particular ways. A tool's underlying approach influence people's overall approach, their decisions on the visualization design, and their willingness to explore alternative designs.

  Q2.A: What kind of benefits do constructive principles bring into people's visualization design process?

  They promote a hands-on, user-driven design approach that enables a deep understanding and control of the visual mappings. In turn, this leads to a better sense of agency (i.e., feeling of control of the process and authorship of the outcomes). Section 4.6 provides more details on this.

  Q2.B: What limitations do they impose?

  Because of the need of repeated interaction steps, a digital constructive visualization tool requires a significant amount of time and effort from the designer. As a consequence, the visualization process might not scale well to large datasets. See section 4.7 for more details.

Q2.C: What strategies could be applied to overcome these limitations?
As investigated in Chapter 5,. we can implement in the tool automated iteration, abstraction, or automated choices. See section 5.6 for more details.

8.2 Research Contributions

This dissertation presented the following main contributions:

- ***Design and implementation of a novel constructive visualization tool***
  I designed and implemented iVoLVER, a web-based pen and touch visual programming environment that supports visualization using a constructive, bottom-up approach. The tool also allows for extraction of data from different types of digital artifacts, supporting visualization in casual and opportunistic settings away from the desk.
- ***Empirical evidence on how opposite approaches to visualization construction affect people's design processes***
  Through a two-part observational study I investigated how people's design approach changes when using two different tools, each implementing a different approach to visualization construction. This study showed how the tools influence: (1) the visualization process, (2) decisions on the visualization design, (3) people's feeling of control and authorship, and (4) their willingness to explore alternative designs.
- ***Characterization of the visualization tools design space in terms of agency and atomicity***
  I further investigated how to design constructive visualization tools that support a user-driven, transparent design process while enabling efficiency and automation. To this end, I ran a series of design workshops that looked at how both visualization experts and novices approach this problem. The findings of this study led to the characterization of the visualization tools design space in terms of *agency* and *atomicity*.

In addition to the main contributions mentioned above, this thesis makes the following secondary contributions:

- The design and implementation of a set of novel widgets that extract data from varied types of digital artifacts.

- A conceptualization of the problem of designing more automated constructive visualization tools.
- A catalogue of example solutions to achieve scalability in constructive visualization tools.

8.3 Summary of Future Work

As detailed in section 7.4 of the Discussion chapter, several questions on digital ConstructiveVis remain open. The opportunities for future work in the area can be summarized as follows:

- Could ConstructiveVis also benefit the visualization process of experts (i.e., people with formal training on visualization design)?
- How could constructive tools look like if not implemented with visual programming? What design goals and principles should be considered for these tools? How would the authoring process of these tools compare to the one of iVoLVER?
- Besides critical reflection, could digital ConstructiveVis also be beneficial for the generation of more or richer insights?
- How could digital ConstructiveVis support visualization of non-tabular datasets (e.g., networks, textual, or spatial data)?
- How could data extraction within visualization tools benefit analysis of data in casual and opportunistic scenarios?

8.4 Closing Remarks

This dissertation has demonstrated that it is possible to incorporate constructive principles in digital visualization tools. It has also deepened our understanding of the benefits and drawbacks of the approach, as well as some potential solutions. The research perspectives for the area of ConstructiveVis in the digital realm seem both promising and exciting. It is my hope that the insights here presented can guide the design of better visualization tools, motivate further research in this area, and assist other researchers in future explorations with digital constructive visualization tools.

References

[1] Russ Abbott. 2004. Emergence, entities, entropy, and binding forces. In *The Agent 2004 Conference on: Social Dynamics: Interaction, Reflexivity, and Emergence*. 17. http://www.necsi.edu/events/iccs/openconf/author/papers/303.pdf

[2] Greg Abram and Lloyd Treinish. 1995. An Extended Data-Flow Architecture for Data Analysis and Visualization. In *Proceedings of the 6th Conference on Visualization '95 (VIS '95)*. IEEE Computer Society, Washington, DC, USA, 263–. http://dl.acm.org/citation.cfm?id=832271.833845

[3] Edith Ackermann. 2001. Piaget's constructivism, Papert's constructionism: What's the difference. *Future of learning group publication* 5, 3 (2001), 438. http://www.sylviastipich.com/wp-content/uploads/2015/04/Coursera-Piaget-_-Papert.pdf

[4] Vladimir Agafonkin. 2013. Simplify.js - A high-performance JavaScript 2D/3D polyline simplification library. (2013). http://mourner.github.io/simplify-js

[5] James F. Allen, Curry I. Guinn, and Eric Horvitz. 1999. Mixed-initiative interaction. *IEEE Intelligent Systems* 14, 5 (1999), 14–23. DOI: http://dx.doi.org/10.1109/5254.796083

[6] Basak Alper, Nathalie Henry Riche, Fanny Chevalier, Jeremy Boy, and Metin Sezgin. 2017. Visualization Literacy at Elementary School. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5485–5497. DOI: http://dx.doi.org/10.1145/3025453.3025877

[7] Daniel Archambault, Tamara Munzner, and David Auber. 2008. GrouseFlocks: Steerable Exploration of Graph Hierarchy Space. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (July 2008), 900–913. DOI: http://dx.doi.org/10.1109/TVCG.2008.34

[8] Sriram Karthik Badam, Eli Fisher, and Niklas Elmqvist. 2015. Munin: A Peer-to-Peer Middleware for Ubiquitous Analytics and Visualization Spaces. *IEEE Transactions on Visualization and Computer Graphics* 21, 2 (Feb. 2015), 215–228. DOI: http://dx.doi.org/10.1109/TVCG.2014.2337337

[9] Ed Baroth and Chris Hartsough. 1995. Visual Object-oriented Programming. In *Visual Object-oriented programming*, Margaret M. Burnett, Adele Goldberg, and Ted G. Lewis (Eds.). Manning Publications Co., Greenwich, CT, USA, Chapter Visual Programming in the Real World, 21–42. http://dl.acm.org/citation.cfm?id=213388.213393

[10] Benjamin B. Bederson and James D. Hollan. 1994. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology (UIST '94)*. ACM, New York, NY, USA, 17–26. DOI: http://dx.doi.org/10.1145/192426.192435

[11] Jaques Bertin. 1999. Readings in Information Visualization. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Chapter Graphics and Graphic Information Processing, 62–65. http://dl.acm.org/citation.cfm?id=300679.300683

[12] Jacques Bertin. 2010. *Semiology of Graphics: Diagrams, Networks, Maps* (1 edition ed.). ESRI Press, Redlands, Calif.

[13] Enrico Bertini, Adam Perer, Catherine Plaisant, and Giuseppe Santucci. 2008. BELIV'08: Beyond Time and Errors: Novel Evaluation Methods for Information Visualization. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems (CHI EA '08)*. ACM, New York, NY, USA, 3913–3916. DOI: http://dx.doi.org/10.1145/1358628.1358955

[14] Enrico Bertini, Catherine Plaisant, and Giuseppe Santucci. 2007. BELIV'06: Beyond Time and Errors; Novel Evaluation Methods for Information Visualization. *interactions* 14, 3 (May 2007), 59–60. DOI: http://dx.doi.org/10.1145/1242421.1242460

[15] Alex Bigelow, Steven Drucker, Danyel Fisher, and Miriah Meyer. 2014. Reflections on How Designers Design with Data. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. ACM, New York, NY, USA, 17–24. DOI: http://dx.doi.org/10.1145/2598153.2598175

[16] Jon Bird and Yvonne Rogers. 2010. The pulse of tidy street: Measuring and publicly displaying domestic electricity consumption. In *Workshop on Energy Awareness and Conservation through Pervasive Applications (Pervasive 2010)*. https://www.idc-online.com/technical_references/pdfs/electrical_engineering/ThepulseofTidystreet.pdf

[17] Elizabeth Ligon Bjork and Robert A Bjork. 2011. Making things hard on yourself, but in a good way: Creating desirable difficulties to enhance learning. *Psychology and the real world: Essays illustrating fundamental contributions to society* 2 (2011), 59–68. https://bjorklab.psych.ucla.edu/wp-content/uploads/sites/13/2016/04/EBjork_RBjork_2011.pdf

[18] Robert A Bjork. 1994. Institutional impediments to effective training. *Learning, remembering, believing: Enhancing human performance* (1994), 295–306. https://bjorklab.psych.ucla.edu/wp-content/uploads/sites/13/2016/07/RBjork_1994b.pdf

[19] Robert A Bjork and Marcia C Linn. 2006. The science of learning and the learning of science. *Aps Observer* 19, 3 (2006). https://www.psychologicalscience.org/observer/the-science-of-learning-and-the-learning-of-science

[20] Alan F. Blackwell. 1996. Metacognitive theories of visual programming: what do we think we are doing?. In *Proceedings 1996 IEEE Symposium on Visual Languages*. 240–246. DOI: http://dx.doi.org/10.1109/VL.1996.545293

[21] Alan F. Blackwell. 2006. *Psychological Issues in End-User Programming*. Springer Netherlands, Dordrecht, 9–30. DOI: http://dx.doi.org/10.1007/1-4020-5386-X_2

[22] André B. Bondi. 2000. Characteristics of Scalability and Their Impact on Performance. In *Proceedings of the 2Nd International Workshop on Software and Performance (WOSP '00)*. ACM, New York, NY, USA, 195–203. DOI: http://dx.doi.org/10.1145/350391.350432

[23] Marat Boshernitsan and Michael Sean Downes. 2004. *Visual programming languages: A survey*. Citeseer. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.2491&rep=rep1&type=pdf

[24] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3; Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec. 2011), 2301–2309. DOI: http://dx.doi.org/10.1109/TVCG.2011.185

[25] Fatma Bouali, Abdelheq Guettala, and Gilles Venturini. 2016. VizAssist: an interactive user assistant for visual data mining. *The Visual Computer* 32, 11 (01 Nov 2016), 1447–1463. DOI: http://dx.doi.org/10.1007/s00371-015-1132-9

[26] Jeremy Boy, Ronald A. Rensink, Enrico Bertini, and Jean-Daniel Fekete. 2014. A Principled Way of Assessing Visualization Literacy. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec 2014), 1963–1972. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346984

[27] Richard E Boyatzis. 1998. *Transforming qualitative information: Thematic analysis and code development*. Sage.

[28] John Brosz, Miguel A. Nacenta, Richard Pusch, Sheelagh Carpendale, and Christophe Hurter. 2013. Transmogrification: Casual Manipulation of Visualizations. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 97–106. DOI: http://dx.doi.org/10.1145/2501988.2502046

[29] Jeffrey Browne, Bongshin Lee, Sheelagh Carpendale, Nathalie Riche, and Timothy Sherwood. 2011. Data Analysis on Interactive Whiteboards Through Sketch-based Interaction. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, New York, NY, USA, 154–157. DOI: http://dx.doi.org/10.1145/2076354.2076383

[30] James C Browne, Jack J Dongarra, Syed I Hyder, Keith Moore, and Peter Newton. 1994. *Visual programming and parallel computing*. University of Tennessee. Computer Science Department.

[31] Margaret M Burnett and Marla J Baker. 1994. A classification system for visual programming languages. *Journal of Visual Languages & Computing* 5, 3 (1994), 287–300. DOI: http://dx.doi.org/10.1006/jvlc.1994.1015

[32] Margaret M. Burnett, Marla J. Baker, Carisa Bohus, Paul Carlson, Sherry Yang, and Pieter Van Zee. 1995. Scaling up visual programming languages. *Computer* 28, 3 (Mar 1995), 45–54. DOI: http://dx.doi.org/10.1109/2.366157

[33] William Buxton and Richard Sniderman. 1980. Iteration in the design of the human-computer interface. In *Proceedings of the 13th Annual Meeting of the Human Factors Association of Canada*, Vol. 7281. 37. https://www.billbuxton.com/iteration.html

[34] Stuart K. Card and Jock Mackinlay. 1997. The structure of the information visualization design space. In *Information Visualization, 1997. Proceedings., IEEE Symposium on*. 92–99. DOI: http://dx.doi.org/10.1109/INFVIS.1997.636792

[35] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. 1999. *Readings in information visualization: using vision to think*. Morgan Kaufmann.

[36] Nicholas Carr. 2015. Nicholas Carr: 'Are we becoming too reliant on computers?' The Guardian. (2015). https://www.theguardian.com/books/2015/jan/17/nicholas-carr-are-we-becoming-too-reliant-on-computers

[37] Liz Carver and Murray Turoff. 2007. Human-computer Interaction: The Human and Computer As a Team in Emergency Management Information Systems. *Commun. ACM* 50, 3 (March 2007), 33–38. DOI: http://dx.doi.org/10.1145/1226736.1226761

[38] Chris Chambers and Chris Scaffidi. 2010. Struggling to Excel: A Field Study of Challenges Faced by Spreadsheet Users. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*. 187–194. DOI: http://dx.doi.org/10.1109/VLHCC.2010.33

[39] Shi-Kuo Chang. 2012. *Visual languages*. Springer Science & Business Media.

[40] William O. Chao, Tamara Munzner, and Michiel van de Panne. 2010. Poster: Rapid pen-centric authoring of improvisational visualizations with napkinvis. *Posters Compendium InfoVis* (2010). http://napkinvis.com/SummaryNapkinVis.pdf

[41] Iskren Ivov Chernev. 2011. MomentJS - Parse, validate, manipulate, and display dates and times in JavaScript. (2011). http://momentjs.com

[42] Ed H. Chi. 2000. A taxonomy of visualization techniques using the data state reference model. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*. 69–75. DOI: http://dx.doi.org/10.1109/INFVIS.2000.885092

[43] Andy Clark and David Chalmers. 1998. The Extended Mind. *Analysis* 58, 1 (1998), 7–19. http://www.jstor.org/stable/3328150

[44] William S Cleveland and Robert McGill. 1984. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *J.* Amer. Statist. Assoc. 79, 387 (1984), 531–554. DOI: http://dx.doi.org/10.2307/2288400

[45] William S. Cleveland and Robert McGill. 1987. Graphical Perception: The Visual Decoding of Quantitative Information on Graphical Displays of Data. *Journal of the Royal Statistical Society. Series A (General)* 150, 3 (1987), pp. 192–229. http://www.jstor.org/stable/2981473

[46] Mathieu Cliche, David Rosenberg, Dhruv Madeka, and Connie Yee. 2017. Scatteract: Automated Extraction of Data from Scatter Plots. In *Machine Learning and Knowledge Discovery in Databases*, Michelangelo Ceci, Jaakko Hollmén, Ljupčo Todorovski, Celine Vens, and Sašo Džeroski (Eds.). Springer International Publishing, Cham, 135–150. DOI: http://dx.doi.org/10.1007/978-3-319-71249-9_9

[47] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. 2009. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Comput. Surv.* 41, 1 (Jan. 2009), 2:1–2:31. DOI: http://dx.doi.org/10.1145/1456650.1456652

[48] Gennaro Costagliola, Mattia De Rosa, Vittorio Fuccella, and Stefano Perna. 2017. Visual languages: A graphical review. *Information Visualization* (2017), 1–16. DOI: http://dx.doi.org/10.1177/1473871617714520

[49] David Coyle, James Moore, Per Ola Kristensson, Paul Fletcher, and Alan Blackwell. 2012. I Did That! Measuring Users' Experience of Agency in Their Own Actions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2025–2034. DOI: http://dx.doi.org/10.1145/2207676.2208350

[50] Cycling '74. 2017. Max Software Tools for Media | Cycling '74. (2017). https://cycling74.com/products/max

[51] Allen Cypher and Daniel Conrad Halbert. 1993. *Watch what I do: programming by demonstration*. MIT press.

[52] Frédéric Dehais, Mickaël Causse, François Vachon, and Sébastien Tremblay. 2012. Cognitive conflict in human–automation interactions: a psychophysiological study. *Applied ergonomics* 43, 3 (2012), 588–595. DOI: http://dx.doi.org/10.1016/j.apergo.2011.09.004

[53] Selan dos Santos and Ken Brodlie. 2004. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics* 28, 3 (2004), 311 – 325. DOI: http://dx.doi.org/10.1016/j.cag.2004.03.013

[54] Edd Dumbill. 2012. *Planning for big data*. O'Reilly Media, Inc.

[55] Cody Dunne and Ben Shneiderman. 2013. Motif Simplification: Improving Network Visualization Readability with Fan, Connector, and Clique Glyphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 3247–3256. DOI: http://dx.doi.org/10.1145/2470654.2466444

[56] Niklas Elmqvist, John Stasko, and Philippas Tsigas. 2008. DataMeadow: A Visual Canvas for Analysis of Large-scale Multivariate Data. *Information Visualization* 7, 1 (March 2008), 18–33. DOI: http://dx.doi.org/10.1145/1391107.1391110

[57] Microsoft Excel. 2017. Create a chart with recommended charts - Excel. (2017). https://support.office.com/en-gb/article/Create-a-chart-with-recommended-charts-cd
-4537-a438-8db20cea84c0 [Online; accessed 16-September-2017].

[58] Jean-Daniel Fekete. 2004. The InfoVis Toolkit. In *IEEE Symposium on Information Visualization*. 167–174. DOI: http://dx.doi.org/10.1109/INFVIS.2004.64

[59] Sara Carolyn Fisher. 1916. The process of generalizing abstraction; and its product, the general concept. *The Psychological Monographs* 21, 2 (1916), i–218. DOI: http://dx.doi.org/10.1037/h0093097

[60] Gapminder Foundation. 2015. Gapminder, a fact-based worldview. (2015). http://www.gapminder.org

[61] Mathias Frisch, Jens Heydekorn, and Raimund Dachselt. 2009. Investigating Multi-touch and Pen Gestures for Diagram Editing on Interactive Surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. ACM, New York, NY, USA, 149–156. DOI: http://dx.doi.org/10.1145/1731903.1731933

[62] Issei Fujishiro, Rika Furuhata, Yoshihiko Ichikawa, and Yuriko Takeshima. 2000. GADGET/IV: A Taxonomic Approach to Semi-Automatic Design of Information Visualization Applications Using Modular Visualization Environment. In *Proceedings of the IEEE Symposium on Information Vizualization 2000 (INFOVIS '00)*. IEEE Computer Society, Washington, DC, USA, 77–. http://dl.acm.org/citation.cfm?id=857190.857689

[63] Dorothy Gabel and Robert Sherwood. 1980. The effect of student manipulation of molecular models on chemistry achievement according to Piagetian level. *Journal of Research in Science Teaching* 17, 1 (1980), 75–81. DOI: http://dx.doi.org/10.1002/tea.3660170112

[64] Tong Gao, Jessica R. Hullman, Eytan Adar, Brent Hecht, and Nicholas Diakopoulos. 2014. NewsViews: An Automated Pipeline for Creating Custom Geovisualizations for News. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3005–3014. DOI: http://dx.doi.org/10.1145/2556288.2557228

[65] Janice D. Gobert and John J. Clement. 1999. Effects of student-generated diagrams versus student-generated summaries on conceptual understanding of causal and dynamic knowledge in plate tectonics. *Journal of Research in Science Teaching* 36, 1 (1999), 39–53. DOI: http://dx.doi.org/10.1002/(SICI)1098-2736(199901)36:1<39::AID-TEA4>3.0.CO;2-I

[66] Eric J. Golin and Steven P. Reiss. 1990. The specification of visual language syntax. *Journal of Visual Languages and Computing* 1, 2 (1990), 141–157. DOI: http://dx.doi.org/10.1016/S1045-926X(05)80013-8

[67] Google. 2017a. Blockly | Google Developers. (2017). https://developers.google.com/blockly/

[68] Google. 2017b. See and use suggested charts and analysis in a spreadsheet. (2017). https://support.google.com/docs/answer/6280499 [Online; accessed 16-September-2017].

[69] David Gotz and Zhen Wen. 2009. Behavior-driven Visualization Recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI '09)*. ACM, New York, NY, USA, 315–324. DOI: http://dx.doi.org/10.1145/1502650.1502695

[70] Lars Grammel, Chris Bennett, Melanie Tory, and Margaret-Anne Storey. 2013. A Survey of Visualization Construction User Interfaces. In *EuroVis - Short Papers*, Mario Hlawitschka and Tino Weinkauf (Eds.). The Eurographics Association. DOI: http://dx.doi.org/10.2312/PE.EuroVisShort.EuroVisShort2013.019-023

[71] Lars Grammel, Melanie Tory, and Margaret-Anne Storey. 2010. How Information Visualization Novices Construct Visualizations. *IEEE Transactions*

on Visualization and Computer Graphics 16, 6 (Nov 2010), 943–952. DOI: http://dx.doi.org/10.1109/TVCG.2010.164

[72] Thomas Green. 1989. Cognitive dimensions of notations. *Proceedings of the fifth conference of the British Computer Society, Human-Computer Interaction Specialist Group on People and computers V* (1989), 443–460. DOI: http://dx.doi.org/citeulike-article-id:634333

[73] John Alexis Guerra-Gomez, Aaron Wilson, Juan Liu, Dan Davies, Peter Jarvis, and Eric Bier. Network Explorer: Design, Implementation, and Real World Deployment of a Large Network Visualization Tool. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, 108–111. DOI: http://dx.doi.org/10.1145/2909132.2909281

[74] Patrick Haggard and Manos Tsakiris. 2009. The Experience of Agency. *Current Directions in Psychological Science* 18, 4 (2009), 242–246. DOI: http://dx.doi.org/10.1111/j.1467-8721.2009.01644.x

[75] Jonathan Harper and Maneesh Agrawala. 2014. Deconstructing and Restyling D3 Visualizations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 253–262. DOI: http://dx.doi.org/10.1145/2642918.2647411

[76] Jonathan Harper and Maneesh Agrawala. 2017. Converting Basic D3 Charts into Reusable Style Templates. *IEEE Transactions on Visualization and Computer Graphics* PP, 99 (2017), 1–1. DOI: http://dx.doi.org/10.1109/TVCG.2017.2659744

[77] Björn Harrtell, Bart van Andel, Primož Pincolič, and Brian Schantz. 2016. JavaScript Topology Suite. (2016). https://github.com/bjornharrtell/jsts

[78] Orit Hazzan. 2003. How Students Attempt to Reduce Abstraction in the Learning of Mathematics and in the Learning of Computer Science. *Computer Science Education* 13, 2 (2003), 95–122. DOI: http://dx.doi.org/10.1076/csed.13.2.95.14202

[79] Christian Heath, Jon Hindmarsh, and Paul Luff. 2010. *Video in qualitative research*. Sage Publications.

[80] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. In *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 203–212. DOI: http://dx.doi.org/10.1145/1753326.1753357

[81] Jeffrey Heer, Stuart K. Card, and James A. Landay. 2005. Prefuse: A Toolkit for Interactive Information Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 421–430. DOI: http://dx.doi.org/10.1145/1054972.1055031

[82] Jeffrey Heer and George Robertson. 2007. Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov 2007), 1240–1247. DOI: http://dx.doi.org/10.1109/TVCG.2007.70539

[83] Daniel D Hils. 1992. Visual languages and computing survey: Data flow visual programming languages. *Journal of Visual Languages & Computing* 3, 1 (March 1992), 69–101. DOI: http://dx.doi.org/10.1016/1045-926X(92)90034-J

[84] Uta Hinrichs and Stefania Forlini. 2017. In defense of sandcastles: research thinking through visualization in DH. In *Proceedings of the conference on Digital Humanities*. International Alliance of Digital Humanities Organizations (ADHO). https://dh2017.adho.org/abstracts/133/133.pdf

[85] Uta Hinrichs, Holly Schmidt, and Sheelagh Carpendale. 2008. EMDialog: Bringing information visualization into the museum. *IEEE transactions on visualization and computer graphics* 14, 6 (2008), 1181–1188. DOI: http://dx.doi.org/10.1109/TVCG.2008.127

[86] Douglas Hofstadter and Emmanuel Sander. 2013. *Surfaces and essences: Analogy as the fuel and fire of thinking*. Basic Books.

[87] John A Hortin. 1980. *Visual Literacy and Visual Thinking.* ERIC.

[88] Dandan Huang, Melanie Tory, Bon Adriel Aseniero, Lyn Bartram, Scott Bateman, Sheelagh Carpendale, Anthony Tang, and Robert Woodbury. 2015. Personal Visualization and Personal Visual Analytics. *Visualization and Computer Graphics, IEEE Transactions on* 21, 3 (March 2015), 420–433. DOI: http://dx.doi.org/10.1109/TVCG.2014.2359887

[89] Weihua Huang and Chew Lim Tan. 2007. A System for Understanding Imaged Infographics and Its Applications. In *Proceedings of the 2007 ACM Symposium on Document Engineering (DocEng '07)*. ACM, New York, NY, USA, 9–18. DOI: http://dx.doi.org/10.1145/1284420.1284427

[90] Darrell Huff. 1993. *How to lie with statistics*. WW Norton & Company.

[91] Jessica Hullman, Eytan Adar, and Priti Shah. 2011. Benefitting InfoVis with Visual Difficulties. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec 2011), 2213–2222. DOI: http://dx.doi.org/10.1109/TVCG.2011.175

[92] Jessica Hullman and Nick Diakopoulos. 2011. Visualization Rhetoric: Framing Effects in Narrative Visualization. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec. 2011), 2231–2240. DOI: http://dx.doi.org/10.1109/TVCG.2011.255

[93] Samuel Huron. 2014. *Constructive Visualization : A token-based paradigm allowing to assemble dynamic visual representation for non-experts*. Theses. Université Paris Sud - Paris XI. https://tel.archives-ouvertes.fr/tel-01126892

[94] Samuel Huron, Sheelagh Carpendale, Jeremy Boy, and Jean-Daniel Fekete. 2016. Using VisKit: A Manual for Running a Constructive Visualization Workshop. In *Pedagogy of Data Visualization Workshop at IEEE VIS 2016*. Baltimore, MD, United States. https://hal.inria.fr/hal-01384388

[95] Samuel Huron, Sheelagh Carpendale, Alice Thudt, Anthony Tang, and Michael Mauerer. 2014. Constructive Visualization. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14)*. ACM, New York, NY, USA, 433–442. DOI: http://dx.doi.org/10.1145/2598510.2598566

[96] Samuel Huron, Pauline Gourlet, Uta Hinrichs, Trevor Hogan, and Yvonne Jansen. 2017. Let's Get Physical: Promoting Data Physicalization in Workshop Formats. In *Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17)*. ACM, New York, NY, USA, 1409–1422. DOI: http://dx.doi.org/10.1145/3064663.3064798

[97] Samuel Huron, Yvonne Jansen, and Sheelagh Carpendale. 2014. Constructing Visual Representations: Investigating the Use of Tangible Tokens. *Visualization and Computer Graphics, IEEE Transactions on* 20, 12 (Dec 2014), 2102–2111. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346292

[98] IBM. 2017. IBM Knowledge Center - Chart Builder layout and terms. (2017). https://www.ibm.com/support/knowledgecenter/en/SSLVMB_20.0.0/com.ibm.spss.statistics.help/chart_create_layout.htm [Online; accessed 16-September-2017].

[99] Itseez Inc. 2015. Open Source Computer Vision Library. https://github.com/itseez/opencv. (2015).

[100] Caleb Jacob and Louis Ameline. 2012. Tooltipster - A flexible and extensible jQuery plugin for modern tooltips. (2012). http://iamceege.github.io/tooltipster

[101] Kelly Pennock David Lantrip Marc Pottier Anne Schur Vern Crow James A. Wise, James J. Thomas. 1995. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proceedings of Visualization 1995 Conference*. 51–58. DOI: http://dx.doi.org/10.1109/INFVIS.1995.528686

[102] Yvonne Jansen and Pierre Dragicevic. 2013. An Interaction Model for Visualizations Beyond The Desktop. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec 2013), 2396–2405. DOI: http://dx.doi.org/10.1109/TVCG.2013.134

[103] Waqas Javed and Niklas Elmqvist. 2013. ExPlates: Spatializing Interactive Analysis to Scaffold Visual Exploration. *Computer Graphics Forum* 32, 3pt4 (2013), 441–450. DOI: http://dx.doi.org/10.1111/cgf.12131

[104] Wesley M. Johnston, J. R. Paul Hanna, and Richard J. Millar. 2004. Advances in Dataflow Programming Languages. *ACM Comput. Surv.* 36, 1 (March 2004), 1–34. DOI: http://dx.doi.org/10.1145/1013208.1013209

[105] Angela Mayhua Jorge Poco and Jeffrey Heer. 2017. Extracting and Retargeting Color Mappings from Bitmap Images of Visualizations. *IEEE Transactions on Visualization and Computer Graphics* PP, 99 (2017), 1–1. DOI: http://dx.doi.org/10.1109/TVCG.2017.2744320

[106] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeong-in Hwang, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. 2017. ChartSense: Interactive Data Extraction from Chart Images. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 6706–6717. DOI: http://dx.doi.org/10.1145/3025453.3025957

[107] Rick Kazman and Jeromy Carrière. 1996. Rapid Prototyping of Information Visualizations Using VANISH. In *Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*. IEEE Computer Society, Washington, DC, USA, 21–. http://dl.acm.org/citation.cfm?id=857187.857619

[108] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. 2006. Challenges in Visual Data Analysis. In *Tenth International Conference on Information Visualisation (IV'06)*. 9–16. DOI: http://dx.doi.org/10.1109/IV.2006.31

[109] John C. Kendrew. 1963. Myoglobin and the Structure of Proteins. *Science* 139, 3561 (1963), 1259–1266. http://www.jstor.org/stable/1711083

[110] Gregor Kiczales. 1996. Beyond the black box: open implementation. *IEEE Software* 13, 1 (Jan 1996), 8, 10–11. DOI: http://dx.doi.org/10.1109/52.476280

[111] Jenny Kitzinger. 1994. The methodology of Focus Groups: the importance of interaction between research participants. *Sociology of Health & Illness* 16, 1 (1994), 103–121. DOI: http://dx.doi.org/10.1111/1467-9566.ep11347023

[112] Jenny Kitzinger. 1995. Qualitative research. Introducing focus groups. *BMJ: British medical journal* 311, 7000 (1995), 299. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2550365

[113] Philipp Koytek, Charles Perin, Jo Vermeulen, Elisabeth André, and Sheelagh Carpendale. 2017. MyBrush: Brushing and Linking with Personal Agency. *IEEE Transactions on Visualization and Computer Graphics* PP, 99 (2017), 1–1. DOI: http://dx.doi.org/10.1109/TVCG.2017.2743859

[114] Richard A Krueger and Mary Anne Casey. 2014. *Focus groups: A practical guide for applied research*. Sage publications. Print.

[115] Interface Ecology Lab. 2015. IdeaMâché, a browser-based expressive web curation system. (2015). http://ideamache.ecologylab.net

[116] Bruno Latour and Couze Venn. 2002. Morality and Technology. *Theory, Culture & Society* 19, 5-6 (2002), 247–260. DOI: http://dx.doi.org/10.1177/026327602761899246

[117] Mathieu Le Goc, Lawrence H. Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building Blocks for Swarm User Interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 97–109. DOI: http://dx.doi.org/10.1145/2984511.2984547

[118] Bongshin Lee, Rubaiat Habib Kazi, and Greg Smith. 2013. SketchStory: Telling More Engaging Stories with Data through Freeform Sketching. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec. 2013), 2416–2425. DOI: http://dx.doi.org/10.1109/TVCG.2013.191

[119] Henry Lieberman. 2000. *Your wish is my command: Giving users the power to instruct their software*.

[120] Ruizhe Liu, Weihua Huang, and Chew Lim Tan. 2007. Extraction of Vectorized Graphical Information from Scientific Chart Images. In *Ninth International Conference on Document Analysis and Recognition, 2007. ICDAR 2007*, Vol. 1. 521–525. DOI: http://dx.doi.org/10.1109/ICDAR.2007.4378764

[121] Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. 2014. A survey on information visualization: recent advances and challenges. *The Visual Computer* 30, 12 (2014), 1373–1393. DOI: http://dx.doi.org/10.1007/s00371-013-0892-3

[122] Gerald L. Lohse, Kevin Biolsi, Neff Walker, and Henry H. Rueter. 1994. A Classification of Visual Representations. *Commun. ACM* 37, 12 (Dec. 1994), 36–49. DOI: http://dx.doi.org/10.1145/198366.198376

[123] Tomislav Lugarić, Zvonimir Pavlić, and Dejan Škvorc. 2016. Object-oriented programming model for synthesis of domain-specific application development environment. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 1162–1167. DOI: http://dx.doi.org/10.1109/MIPRO.2016.7522315

[124] Sean Lynch, Miguel A. Nacenta, and Sheelagh Carpendale. 2011. ToCoPlay: Graphical Multi-touch Interaction for Composing and Playing Music. In *Human-Computer Interaction - INTERACT 2011*, Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler (Eds.). Number 6948 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 306–322. http://link.springer.com/chapter/10.1007/978-3-642-23765-2_22

[125] Kwan-Liu Ma. 1999. Image Graphs - a Novel Approach to Visual Data Exploration. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years (VIS '99)*. IEEE Computer Society Press, Los Alamitos, CA, USA, 81–88. http://dl.acm.org/citation.cfm?id=319351.319360

[126] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show Me: Automatic Presentation for Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov 2007), 1137–1144. DOI: http://dx.doi.org/10.1109/TVCG.2007.70594

[127] Redacción Madrid. 2015. Un nuevo gráfico 'manipulado' sobre el paro se cuela en TVE. *La Vanguardia* (Jan. 2015). http://www.lavanguardia.com/television/programas/20150123/54424726878/grafico-manipulacion-tve-paro.html

[128] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The Scratch Programming Language and Environment. *Trans. Comput. Educ.* 10, 4, Article 16 (Nov. 2010), 15 pages. DOI: http://dx.doi.org/10.1145/1868358.1868363

[129] Adam V. Maltese, Joseph A. Harsh, and Dubravka Svetina. 2015. Data visualization literacy: investigating data interpretation along the novice–expert continuum. *Journal of College Science Teaching* 45, 1 (2015), 84–90. http://www.jstor.org/stable/43631889

[130] John P Manning. 2005. Rediscovering Froebel: A call to re-examine his life & gifts. *Early Childhood Education Journal* 32, 6 (2005), 371–376. DOI: http://dx.doi.org/10.1007/s10643-005-0004-8

[131] MathWorks. 2012. Simulink: Simulation and Model-Based Design. (2012). https://uk.mathworks.com/products/simulink.htmlhttp://www.mathworks.com/products/simulink

[132] Bryan McDonnel and Niklas Elmqvist. 2009. Towards Utilizing GPUs in Information Visualization: A Model and Implementation of Image-Space Operations. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov 2009), 1105–1112. DOI: http://dx.doi.org/10.1109/TVCG.2009.191

[133] Marshall McLuhan. 2002. *Understanding Media - The Extensions of Man* (critical edition edition ed.). Gingko Press, Inc, Corte Madera, CA.

[134] Emily McReynolds, Sarah Hubbard, Timothy Lau, Aditya Saraf, Maya Cakmak, and Franziska Roesner. 2017. Toys That Listen: A Study of Parents, Children, and Internet-Connected Toys. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5197–5207. DOI: http://dx.doi.org/10.1145/3025453.3025735

[135] Honghui Mei, Yuxin Ma, Yating Wei, and Wei Chen. 2017. The design space of construction tools for information visualization: A survey. *Journal of Visual Languages & Computing* (2017). DOI: http://dx.doi.org/https://doi.org/10.1016/j.jvlc.2017.10.001

[136] Gonzalo Gabriel Méndez. 2016. Tools for opportunistic information visualization: Visual analysis with non-traditional data sources. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 250–251. DOI: http://dx.doi.org/10.1109/VLHCC.2016.7739701

[137] Gonzalo Gabriel Méndez and Miguel A. Nacenta. 2016. Opportunistic visualization with iVoLVER. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 234–235. DOI: http://dx.doi.org/10.1109/VLHCC.2016.7739693

[138] Gonzalo Gabriel Méndez, Miguel A. Nacenta, and Sebastien Vandenheste. 2016. iVoLVER: Interactive Visual Language for Visualization Extraction and Reconstruction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4073–4085. DOI: http://dx.doi.org/10.1145/2858036.2858435

[139] Luana Micallef, Pierre Dragicevic, and Jean-Daniel Fekete. 2012. Assessing the Effect of Visualizations on Bayesian Reasoning through Crowdsourcing. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (Dec 2012), 2536–2545. DOI: http://dx.doi.org/10.1109/TVCG.2012.199

[140] Thomas Mühlbacher, Harald Piringer, Samuel Gratzl, Michael Sedlmair, and Marc Streit. 2014. Opening the Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec 2014), 1643–1652. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346578

[141] Jurriaan D Mulder, Jarke J Van Wijk, and Robert Van Liere. 1999. A survey of computational steering environments. *Future generation computer systems* 15, 1 (1999), 119–129. DOI: http://dx.doi.org/10.1016/s0167-739x(98)00047-8

[142] Michael J. Muller and Sarah Kuhn. 1993. Participatory Design. *Commun. ACM* 36, 6 (June 1993), 24–28. DOI: http://dx.doi.org/10.1145/153571.255960

[143] Tamara Munzner. 2014. *Visualization Analysis and Design* (har/psc edition ed.). A K Peters/CRC Press, Boca Raton. http://www.cs.ubc.ca/~tmm/vadbook

[144] Brad A. Myers. 1990. Taxonomies of visual programming and program visualization. *Journal of Visual Languages & Computing* 1, 1 (March 1990), 97–123. DOI: http://dx.doi.org/10.1016/S1045-926X(05)80036-9

[145] Brad A. Myers. 1992. Demonstrational Interfaces: A Step Beyond Direct Manipulation. *Computer* 25, 8 (1992), 61–73. DOI: http://dx.doi.org/10.1109/2.153286

[146] Bonnie A Nardi. 1993. *A small matter of programming: perspectives on end user computing*. MIT press. Print.

[147] Jakob Nielsen. 1993. Iterative user-interface design. *Computer* 26, 11 (1993), 32–41. DOI: http://dx.doi.org/10.1109/2.241424

[148] Bettina Nissen and John Bowers. 2015. Data-Things: Digital Fabrication Situated Within Participatory Data Translation Activities. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2467–2476. DOI: http://dx.doi.org/10.1145/2702123.2702245

[149] Anshul Vikram Pandey, Katharina Rall, Margaret L. Satterthwaite, Oded Nov, and Enrico Bertini. 2015. How Deceptive Are Deceptive Visualizations?: An Empirical Analysis of Common Distortion Techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1469–1478. DOI: http://dx.doi.org/10.1145/2702123.2702608

[150] Kostas Pantazos and Søren Lauesen. 2012. Constructing Visualizations with InfoVis Tools-An Evaluation from a user Perspective. *GRAPP/IVAPP* 2012 (2012), 731–736. http://www.itu.dk/people/slauesen/S-EHR/Constructing_Visualizations.pdf

[151] Seymour Papert. 1993. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.

[152] Seymour Papert and Idit Harel. 1991. Situating Constructionism. In *Constructionism*. Ablex Publishing Corporation. http://namodemello.com.br/pdf/tendencias/situatingconstrutivism.pdf

[153] Raja Parasuraman and Dietrich H Manzey. 2010. Complacency and bias in human use of automation: An attentional integration. *Human factors* 52, 3 (2010), 381–410. DOI: http://dx.doi.org/10.1177/0018720810376055

[154] Raja Parasuraman, Robert Molloy, and Indramani L Singh. 1993. Performance consequences of automation-induced 'complacency'. *The International Journal of Aviation Psychology* 3, 1 (1993), 1–23. DOI: http://dx.doi.org/10.1207/s15327108ijap0301_1

[155] Charles Perin, Pierre Dragicevic, and Jean-Daniel Fekete. 2014. Revisiting Bertin Matrices: New Interactions for Crafting Tabular Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec 2014), 2082–2091. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346279

[156] Daniel B Perry, Bill Howe, Alicia MF Key, and Cecilia Aragon. 2013. VizDeck: Streamlining exploratory visual analytics of scientific data. (2013). http://hdl.handle.net/2142/36044

[157] Jean Piaget. 2013. *Child's Conception of Space: Selected Works*. Vol. 4. Routledge.

[158] Catherine Plaisant. 2004. The Challenge of Information Visualization Evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 109–116. DOI: http://dx.doi.org/10.1145/989863.989880

[159] Jorge Poco and Jeffrey Heer. 2017. Reverse-Engineering Visualizations: Recovering Visual Encodings from Chart Images. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 353–363. DOI: http://dx.doi.org/10.1111/cgf.13193

[160] Zachary Pousman, John Stasko, and Michael Mateas. 2007. Casual Information Visualization: Depictions of Data in Everyday Life. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1145–1152. DOI: http://dx.doi.org/10.1109/TVCG.2007.70541

[161] Vy Lam Purvi Saraiya, Chris North and Karen A. Duca. 2006. An Insight-Based Longitudinal Study of Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 12, 6 (Nov 2006), 1511–1522. DOI: http://dx.doi.org/10.1109/TVCG.2006.85

[162] Willard Van Orman Quine. 2016. *Levels of Abstraction (1972)*. Palgrave Macmillan UK, London, 12–20. DOI: http://dx.doi.org/10.1057/9781137472519_3

[163] RapidMiner. 2016. Data Science Platform - RapidMiner. (2016). https://rapidminer.com

[164] Casey Reas and Benjamin Fry. 2003. Processing: A Learning Environment for Creating Interactive Web Graphics. In *ACM SIGGRAPH 2003 Web Graphics (SIGGRAPH '03)*. ACM, New York, NY, USA, 1–1. DOI: http://dx.doi.org/10.1145/965333.965356

[165] Donghao Ren, T. Hollerer, and Xiaoru Yuan. 2014. iVisDesigner: Expressive Interactive Design of Information Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec. 2014), 2092–2101. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346291

[166] Mitchel Resnick. 2007. All I Really Need to Know (About Creative Thinking) I Learned (by Studying How Children Learn) in Kindergarten. In *Proceedings of the 6th ACM SIGCHI Conference on Creativity &Amp; Cognition (C&C '07)*. ACM, New York, NY, USA, 1–6. DOI: http://dx.doi.org/10.1145/1254960.1254961

[167] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: Programming for All. *Commun. ACM* 52, 11 (Nov. 2009), 60–67. DOI: http://dx.doi.org/10.1145/1592761.1592779

[168] Jonathan C Roberts, Christopher Headleand, David Perkins, and Panagiotis D Ritsos. 2015. Personal Visualization for Learning. *Personal Visualization: Exploring Data in Everyday Life Workshop, IEEE Conference on Visualization, Chicago, IL, USA* October (2015). http://vig-chester.org/publications/2015/09/06/publications

[169] Jonathan C. Roberts, Chris Headleand, and Panagiotis D. Ritsos. 2016. Sketching Designs Using the Five Design-Sheet Methodology. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (Jan 2016), 419–428. DOI: http://dx.doi.org/10.1109/TVCG.2015.2467271

[170] Jonathan C. Roberts, Panagiotis D. Ritsos, Sriram Karthik Badam, Dominique Brodbeck, Jessie Kennedy, and Niklas Elmqvist. 2014. Visualization beyond

the Desktop–the Next Big Thing. *IEEE Computer Graphics and Applications* 34, 6 (Nov 2014), 26–34. DOI: http://dx.doi.org/10.1109/MCG.2014.82

[171] George G. Robertson, Jock D. Mackinlay, and Stuart K. Card. 1991. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*. ACM, New York, NY, USA, 189–194. DOI: http://dx.doi.org/10.1145/108844.108883

[172] Ankit Rohatgi. 2015. WebPlotDigitizer: Web-based Tool to Extract Data from Plots, Images, and Maps. (2015). https://automeris.io/WebPlotDigitize

[173] Ola Rosling, Anna Rosling, and Rosling Hans. 2015. Gapminder: Unveiling the beauty of statistics for a fact based world view. (2015). DOI: http://dx.doi.org/10.5860/choice.48-3052

[174] Greg Ross and Matthew Chalmers. 2003. A Visual Workspace for Hybrid Multidimensional Scaling Algorithms. In *Proceedings of the Ninth Annual IEEE Conference on Information Visualization (INFOVIS'03)*. IEEE Computer Society, Washington, DC, USA, 91–96. DOI: http://dx.doi.org/10.1109/INFVIS.2003.1249013

[175] Jeffrey M. Rzeszotarski and Aniket Kittur. 2014. Kinetica: Naturalistic Multi-touch Data Visualization. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 897–906. DOI: http://dx.doi.org/10.1145/2556288.2557231

[176] Bahador Saket, Hannah Kim, Eli T. Brown, and Alex Endert. 2017. Visualization by Demonstration: An Interaction Paradigm for Visual Data Exploration. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 331–340. DOI: http://dx.doi.org/10.1109/TVCG.2016.2598839

[177] Arvind Satyanarayan and Jeffrey Heer. 2014. Lyra: An Interactive Visualization Design Environment. *Computer Graphics Forum* 33, 3 (June 2014), 351–360. DOI: http://dx.doi.org/10.1111/cgf.12391

[178] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. 2011. ReVision: Automated Classification, Analysis and Redesign of Chart Images. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 393–402. DOI: http://dx.doi.org/10.1145/2047196.2047247

[179] Sebastian Schmidt, Miguel A. Nacenta, Raimund Dachselt, and Sheelagh Carpendale. 2010. A Set of Multi-touch Graph Interaction Techniques. In *ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. ACM, New York, NY, USA, 113–116. DOI: http://dx.doi.org/10.1145/1936652.1936673

[180] Alexander Schmitz, Chris Thoburn, and Jorik Tangelder. 2014. Hammer.js - A JavaScript library for gesture support. (2014). http://hammerjs.github.io

[181] Ben Schneiderman and Catherine Plaisant. 1998. *Designing the user interface*. Don Mills, Ontario: Addison Wesley.

[182] Jean Scholtz. 2006. Beyond Usability: Evaluation Aspects of Visual Analytic Environments. In *2006 IEEE Symposium On Visual Analytics Science And Technology*. 145–150. DOI: http://dx.doi.org/10.1109/VAST.2006.261416

[183] Konrad J. Schönborn and Trevor R. Anderson. 2006. The importance of visual literacy in the education of biochemists. *Biochemistry and Molecular Biology Education* 34, 2 (2006), 94–102. DOI: http://dx.doi.org/10.1002/bmb.2006.49403402094

[184] Douglas Schuler and Aki Namioka. 1993. *Participatory design: Principles and practices*. CRC Press.

[185] Edward Segel and Jeffrey Heer. 2010. Narrative visualization: Telling stories with data. *IEEE transactions on visualization and computer graphics* 16, 6 (2010), 1139–1148. DOI: http://dx.doi.org/10.1109/TVCG.2010.179

[186] Matthew O. Ward Shiping Huang and Elke A. Rundensteiner. 2005. Exploration of dimensionality reduction for text visualization. In *Coordinated and Multiple Views in Exploratory Visualization (CMV'05)*. 63–74. DOI: http://dx.doi.org/10.1109/CMV.2005.8

[187] Ben Shneiderman. 1993. Direct manipulation: A Step Beyond Programming Languages. *Sparks of innovation in human-computer interaction* (1993), 17. DOI: http://dx.doi.org/10.1109/MC.1983.1654471

[188] Artur Šilić and Bojana Dalbelo Bašić. 2010. Visualization of text streams: A survey. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 31–43. DOI: http://dx.doi.org/10.1007/978-3-642-15390-7_4

[189] Burrhus Frederic Skinner. 1977. Why I Am Not a Cognitive Psychologist. *Behaviorism* 5, 2 (1977), 1–10. http://www.jstor.org/stable/27758892

[190] Randall B. Smith. 1987. Experiences with the Alternate Reality Kit: An Example of the Tension Between Literalism and Magic. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface (CHI '87)*. ACM, New York, NY, USA, 61–67. DOI: http://dx.doi.org/10.1145/29933.30861

[191] Michael E Smoot, Keiichiro Ono, Johannes Ruscheinski, Peng-Liang Wang, and Trey Ideker. 2010. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics* 27, 3 (2010), 431–432. DOI: http://dx.doi.org/10.1093/bioinformatics/btq675

[192] Alan Snyder. 1986. Encapsulation and Inheritance in Object-oriented Programming Languages. In *Conference Proceedings on Object-oriented Programming Systems, Languages and Applications (OOPSLA '86)*. ACM, New York, NY, USA, 38–45. DOI: http://dx.doi.org/10.1145/28697.28702

[193] Tableau Software. 2017a. Business Intelligence and Analytics. (2017). https://www.tableau.com

[194] Tableau Software. 2017b. Columns and Rows Shelves. (2017). http://onlinehelp.tableau.com/current/pro/desktop/en-us/buildmanual_shelves_columnrow.html [Online; accessed 19-September-2017].

[195] Tableau Software. 2017c. Data Pane. (2017). http://onlinehelp.tableau.com/current/pro/desktop/en-us/environ_workspace_datawindow.html [Online; accessed 19-September-2017].

[196] Tableau Software. 2017d. Marks Card. (2017). http://onlinehelp.tableau.com/current/pro/desktop/en-us/buildmanual_shelves_marks.html [Online; accessed 19-September-2017].

[197] Robert Spence. 2001. *Information visualization*. Vol. 1. Springer.

[198] George Stiny. 1980. Kindergarten grammars: designing with Froebel's building gifts. *Environment and Planning B: Planning and Design* 7, 4 (1980), 409–462. DOI: http://dx.doi.org/10.1068/b070409

[199] Lucy Suchman. 2007. Agencies in technology design: Feminist reconfigurations. (2007). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.548.7968&rep=rep1&type=pdf

[200] Guo-Dao Sun, Ying-Cai Wu, Rong-Hua Liang, and Shi-Xia Liu. 2013. A Survey of Visual Analytics Techniques and Applications: State-of-the-Art Research and Future Challenges. *Journal of Computer Science and Technology* 28, 5 (Sept. 2013), 852–867. DOI: http://dx.doi.org/10.1007/s11390-013-1383-8

[201] Steven L. Tanimoto. 1990. VIVA: A visual language for image processing. *Journal of Visual Languages & Computing* 1, 2 (1990), 127–139. DOI: http://dx.doi.org/10.1016/S1045-926X(05)80012-6

[202] Alex S Taylor, Siân Lindley, Tim Regan, David Sweeney, Vasillis Vlachokyriakos, Lillie Grainger, and Jessica Lingel. 2015. Data-in-Place: Thinking Through the Relations Between Data and Community. In *CHI '15, Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM âĂŞ Association for Computing Machinery, New York, NY, USA, 2863–2872. DOI: http://dx.doi.org/10.1145/2702123.2702558

[203] Lauritz Thamsen, Thomas Renner, Marvin Byfeld, Markus Paeschke, Daniel Schroder, and Felix Bohm. 2016. Visually programming dataflows for distributed data analytics. *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016* (2016), 2276–2285. DOI: http://dx.doi.org/10.1109/BigData.2016.7840860

[204] Alice Thudt, Dominikus Baur, Samuel Huron, and Sheelagh Carpendale. 2016. Visual mementos: Reflecting memories with personal data. *IEEE transactions on visualization and computer graphics* 22, 1 (2016), 369–378. DOI: http://dx.doi.org/10.1109/TVCG.2015.2467831

[205] Matthew Tobiasz, Petra Isenberg, and Sheelagh Carpendale. 2009. Lark: Coordinating Co-located Collaboration with Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov 2009), 1065–1072. DOI: http://dx.doi.org/10.1109/TVCG.2009.162

[206] Melanie Tory and Torsten Möller. 2004. Rethinking Visualization: A High-Level Taxonomy. In *IEEE Symposium on Information Visualization*. 151–158. DOI: http://dx.doi.org/10.1109/INFVIS.2004.59

[207] Jeffrey Travis and Jim Kring. 2007. *LabVIEW for everyone: graphical programming made easy and fun*. Prentice-Hall.

[208] Bas Tummers, J Van der Lann, and Kees Huyser. 2006. DataThief III. (2006). http://datathief.org

[209] Barbara Tversky, Julie Bauer Morrison, and Mirelle Betancourt. 2002. Animation: can it facilitate? *International Journal of Human-Computer Studies* 57, 4 (2002), 247 – 262. DOI: http://dx.doi.org/https://doi.org/10.1006/ijhc.2002.1017

[210] Nina Valkanova, Sergi Jorda, and Andrew Vande Moere. 2015. Public visualization displays of citizen data: design, impact and implications. *International Journal of Human-Computer Studies* 81 (2015), 4–16. DOI: http://dx.doi.org/10.1016/j.ijhcs.2015.02.005

[211] Sebastien Vandenheste. 2015. iVoLVR: Pre-commercialization, Usability Improvements and Future Productization. (2015). Master's Thesis. University of St Andrews. St Andrews, United Kingdom.

[212] Bret Victor. 2013. Drawing Dynamic Visualizations. Recorded at the Stanford HCI seminar. (2013). https://vimeo.com/66085662

[213] Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. 2007a. ManyEyes: a Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1121–1128. DOI: http://dx.doi.org/10.1109/TVCG.2007.70577

[214] Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. 2007b. ManyEyes: a Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1121–1128. DOI: http://dx.doi.org/10.1109/TVCG.2007.70577

[215] Robert Voigt. 2002. An extended scatterplot matrix and case studies in information visualization. (2002). http://old.vrvis.at/via/resources/DA-RVoigt/DA.pdf Master's thesis, Hochschule Magdeburg-Stendal.

[216] Paul Vorbach. 2015. Tesseract4Java - Java GUI for the Tesseract OCR engine. https://github.com/tesseract4java. (2015).

[217] Jagoda Walny, Samuel Huron, and Sheelagh Carpendale. 2015. An Exploratory Study of Data Sketching for Visual Representation. *Computer Graphics Forum* 34, 3 (2015), 231–240. DOI: http://dx.doi.org/10.1111/cgf.12635

[218] Colin Ware. 2010. *Visual thinking: For design*. Morgan Kaufmann.

[219] Colin Ware. 2012. *Information visualization: perception for design*. Elsevier.

[220] Kirsten N. Whitley. 1997. Visual Programming Languages and the Empirical Evidence For and Against. *Journal of Visual Languages & Computing* 8, 1 (1997), 109–142. DOI: http://dx.doi.org/10.1006/jvlc.1996.0030

[221] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (Jan 2016), 649–658. DOI: http://dx.doi.org/10.1109/TVCG.2015.2467191

[222] William Wright, David Schroh, Pascale Proulx, Alex Skaburskis, and Brian Cort. 2006. The Sandbox for Analysis: Concepts and Methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 801–810. DOI: http://dx.doi.org/10.1145/1124772.1124890

[223] Tiffany Wun, Jennifer Payne, Samuel Huron, and Sheelagh Carpendale. 2016. Comparing Bar Chart Authoring with Microsoft Excel and Tangible Tiles. *Computer Graphics Forum* (2016). DOI: http://dx.doi.org/10.1111/cgf.12887

[224] Jun Xie, Holger Winnemöller, Wilmot Li, and Stephen Schiller. 2017. Interactive Vectorization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 6695–6705. DOI: http://dx.doi.org/10.1145/3025453.3025872

[225] Guo Yu and Alan Blackwell. 2017. Effects of Timing on Users' Agency during Mixed-Initiative Interaction. (2017). http://hci2017.bcs.org/wp-content/uploads/21.pdf

[226] Juriy Zaytsev, Stefan Kienzle, and Andrea Bogazzi. 2012. Fabric.js - A JavaScript Canvas Library. (2012). http://fabricjs.com

Appendix A

Ethical Approval for Research

A.1 Ethical Approval Letter for Interviews

University of St Andrews
Scotland's first university – 1413

University Teaching and Research Ethics Committee Sub-committee

22nd July 2015
Gonzalo Mendez
School of Computer Science

Ethics Reference No: *Please quote this ref on all correspondence*	CS11702
Project Title:	Supporting data analysis and exploration through iVoLVR
Researchers Name(s):	Gonzalo Mendez
Supervisor(s):	Dr Miguel Nacenta

Thank you for submitting your application which was considered at the Computer Science School Ethics Committee meeting on the 21/07/2015. The following documents were reviewed:

1. Ethical Application Form 15/07/2015
2. Participant Information Sheet 15/07/2015
3. Consent Form 15/07/2015
4. Debriefing Form 15/07/2015

The University Teaching and Research Ethics Committee (UTREC) approve this study from an ethical point of view. Please note that where approval is given by a School Ethics Committee that committee is part of UTREC and is delegated to act for UTREC.

Approval is given for three years. Projects, which have not commenced within two years of original approval, must be re-submitted to your School Ethics Committee.

You must inform your School Ethics Committee when the research has been completed. If you are unable to complete your research within the 3 three year validation period, you will be required to write to your School Ethics Committee and to UTREC (where approval was given by UTREC) to request an extension or you will need to re-apply.

Any serious adverse events or significant change which occurs in connection with this study and/or which may alter its ethical consideration must be reported immediately to the School Ethics Committee, and an Ethical Amendment Form submitted where appropriate.

Approval is given on the understanding that the 'Guidelines for Ethical Research Practice' https://www.st-andrews.ac.uk/utrec/guidelines/ are adhered to.

Yours sincerely

P.P.

UNIVERSITY OF ST ANDREWS SCHOOL OF COMPUTER SCIENCE

Convenor of the School Ethics Committee

Ccs Supervisor
School Ethics Committee

A.2 Ethical Approval Letter for Observational In-classroom Pilot

University of St Andrews
Scotland's first university – 1413

University Teaching and Research Ethics Committee Sub-committee

3rd February 2016
Gonzalo Mendez
School of Computer Science

Ethics Reference No: *Please quote this ref on all correspondence*	CS11923
Project Title:	Studying the construction of visual encodings with iVoLVER and Tableau
Researchers Name(s):	Gonzalo Mendez and Uta Hinrichs
Supervisor(s):	Dr Miguel Nacenta

Thank you for submitting your application which was considered at the Computer Science School Ethics Committee meeting on the 03/02/2016. The following documents were reviewed:

1. Ethical Application Form 02/02/2016
2. Participant Information Sheet 02/02/2016
3. Consent Form 15/01/2016
4. Debriefing Form 15/01/2016
5. Demographic Information 15/01/2016
6. Group Discussion – Sample Questions 15/01/2016
7. Questionnaire – Sample Questions 15/01/2016
8. Quiz – Sample Questions 15/01/2016

The University Teaching and Research Ethics Committee (UTREC) approve this study from an ethical point of view. Please note that where approval is given by a School Ethics Committee that committee is part of UTREC and is delegated to act for UTREC.

Approval is given for three years. Projects, which have not commenced within two years of original approval, must be re-submitted to your School Ethics Committee.

You must inform your School Ethics Committee when the research has been completed. If you are unable to complete your research within the 3 three year validation period, you will be required to write to your School Ethics Committee and to UTREC (where approval was given by UTREC) to request an extension or you will need to re-apply.

Any serious adverse events or significant change which occurs in connection with this study and/or which may alter its ethical consideration must be reported immediately to the School Ethics Committee, and an Ethical Amendment Form submitted where appropriate.

Approval is given on the understanding that the 'Guidelines for Ethical Research Practice' https://www.st-andrews.ac.uk/utrec/guidelines/ are adhered to.

Yours sincerely

UNIVERSITY OF ST ANDREWS SCHOOL OF COMPUTER SCIENCE

pp.

Convenor of the School Ethics Committee

Ccs Supervisor
School Ethics Committee

A.3 Ethical Approval Letter for Observational Lab Study

University of St Andrews
Scotland's first university – 1413

University Teaching and Research Ethics Committee Sub-committee

Project Title	Studying the construction of visual encodings with iVoLVER and Tableau
Researchers Name(s)	Gonzalo Gabriel Méndez, Dr. Uta Hinrichs
Supervisor(s)	Dr. Miguel Nacenta
Department/Unit	School of Computer Science
Ethical Approval Code (Approval allocated to Original Application)	CS11923
Original Application Approval Date	3rd February 2016
Amendment Application Approval	3rd August 2016

Ethical Amendment Approval

Thank you for submitting your amendment application which was considered at the Computer Science School Ethics Committee meeting on the 3rd August 2016. The following documents were reviewed:

1. Ethical Amendment Application Form 15/07/2016
2. Advertisement 15/07/2016
3. Participant Information Sheet 15/07/2016
4. Consent Form 15/07/2016
5. Sample Questions 15/07/2016

The University Teaching and Research Ethics Committee (UTREC) approves this study from an ethical point of view. Please note that where approval is given by a School Ethics Committee that committee is part of UTREC and is delegated to act for UTREC.

Approval is given for three years from the original application only. Ethical Amendments do not extend this period but give permission to an amendment to the original approval research proposal only. If you are unable to complete your research within the original 3 three year validation period, you will be required to write to your School Ethics Committee and to UTREC (where approval was given by UTREC) to request an extension or you will need to re-apply. You must inform your School Ethics Committee when the research has been completed.

Any serious adverse events or significant change which occurs in connection with this study and/or which may alter its ethical consideration, must be reported immediately to the School Ethics Committee, and an Ethical Amendment Form submitted where appropriate.

Approval is given on the understanding that the 'Guidelines for Ethical Research Practice' (http://www.st-andrews.ac.uk/media/UTRECguidelines%20Feb%2008.pdf) are adhered to.

Yours sincerely

P.P.

Convenor of the School Ethics Committee
Ccs Supervisor
School Ethics Committee

A.4 Ethical Approval Letter for Design Study

University of St Andrews
Scotland's first university – 1413

University Teaching and Research Ethics Committee Sub-committee

16/11/2016
Gonzalo Gabriel Mendez and Uta Hinrichs
School of Computer Science

Ethics Reference No: *Please quote this ref on all correspondence*	CS12448
Project Title:	Designing Language Elements and Interactions to Automate iVoLVER Visualizations
Researchers Name(s):	Gonzalo Gabriel Mendez and Uta Hinrichs
Supervisor(s):	Miguel Nacenta

Thank you for submitting your application which was considered at the Computer Science School Ethics Committee meeting on the 16/11/2016. The following documents were reviewed:

1. Ethical Application Form 16/11/2016
2. Participant Information Sheet 16/11/2016
3. Consent Form 16/11/2016
4. Debriefing Form 16/11/2016

The University Teaching and Research Ethics Committee (UTREC) approve this study from an ethical point of view. Please note that where approval is given by a School Ethics Committee that committee is part of UTREC and is delegated to act for UTREC.

Approval is given for three years. Projects, which have not commenced within two years of original approval, must be re-submitted to your School Ethics Committee.

You must inform your School Ethics Committee when the research has been completed. If you are unable to complete your research within the 3 three year validation period, you will be required to write to your School Ethics Committee and to UTREC (where approval was given by UTREC) to request an extension or you will need to re-apply.

Any serious adverse events or significant change which occurs in connection with this study and/or which may alter its ethical consideration must be reported immediately to the School Ethics Committee, and an Ethical Amendment Form submitted where appropriate.

Approval is given on the understanding that the 'Guidelines for Ethical Research Practice' https://www.st-andrews.ac.uk/utrec/guidelines/ are adhered to.

Yours sincerely

UNIVERSITY OF ST. ANDREWS SCHOOL OF COMPUTER SCIENCE

Convenor of the School Ethics Committee

Ccs Supervisor
School Ethics Committee

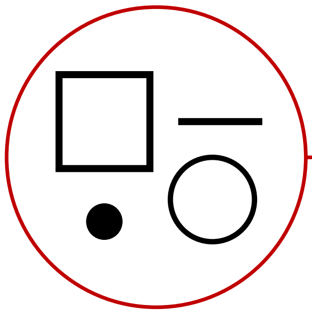
Appendix B

Presentations and Tool Cheatsheets for Studies with iVoLVER and Tableau

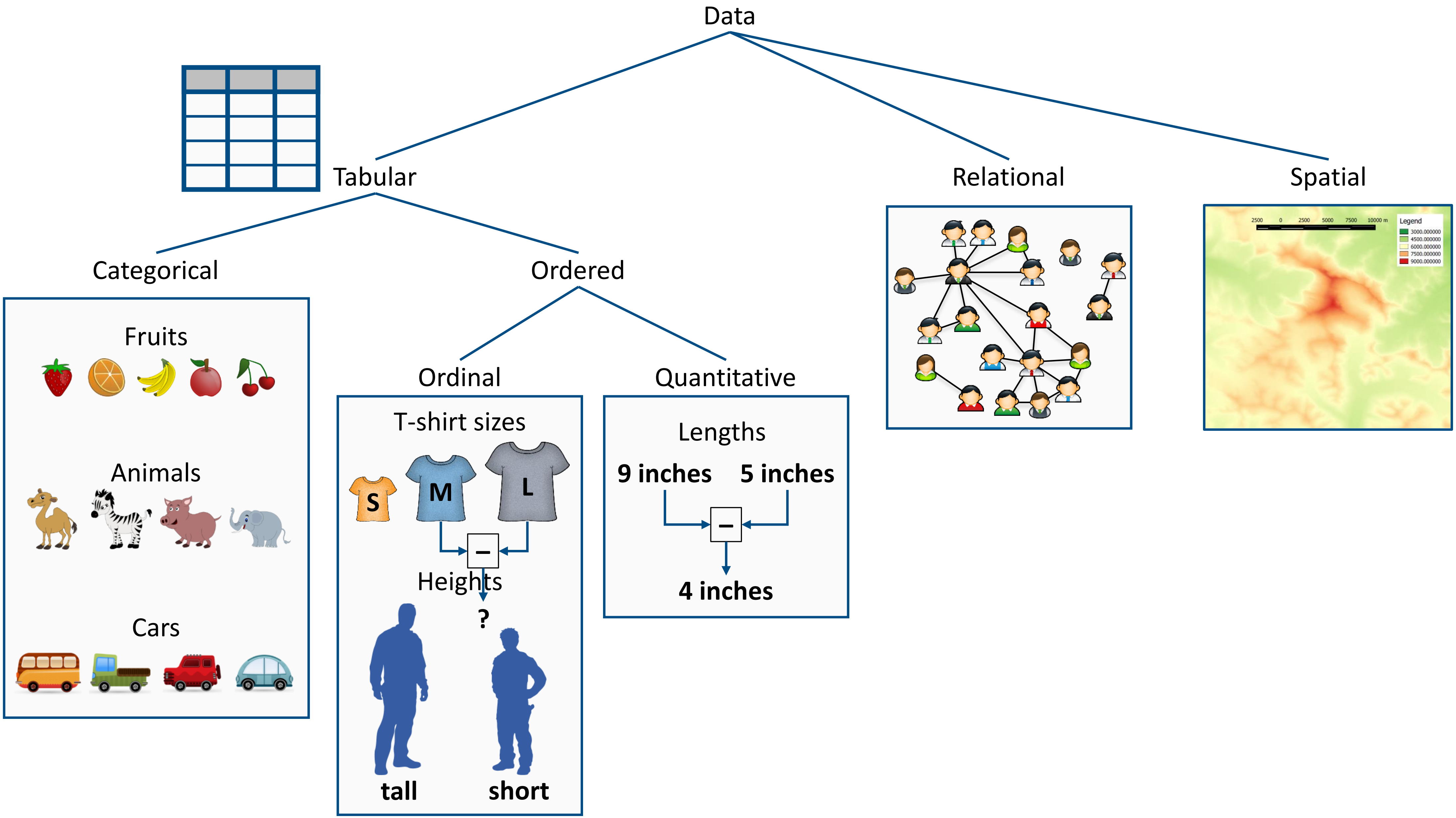
B.1 Introduction to Infovis: Slides from the Video

Information Visualization

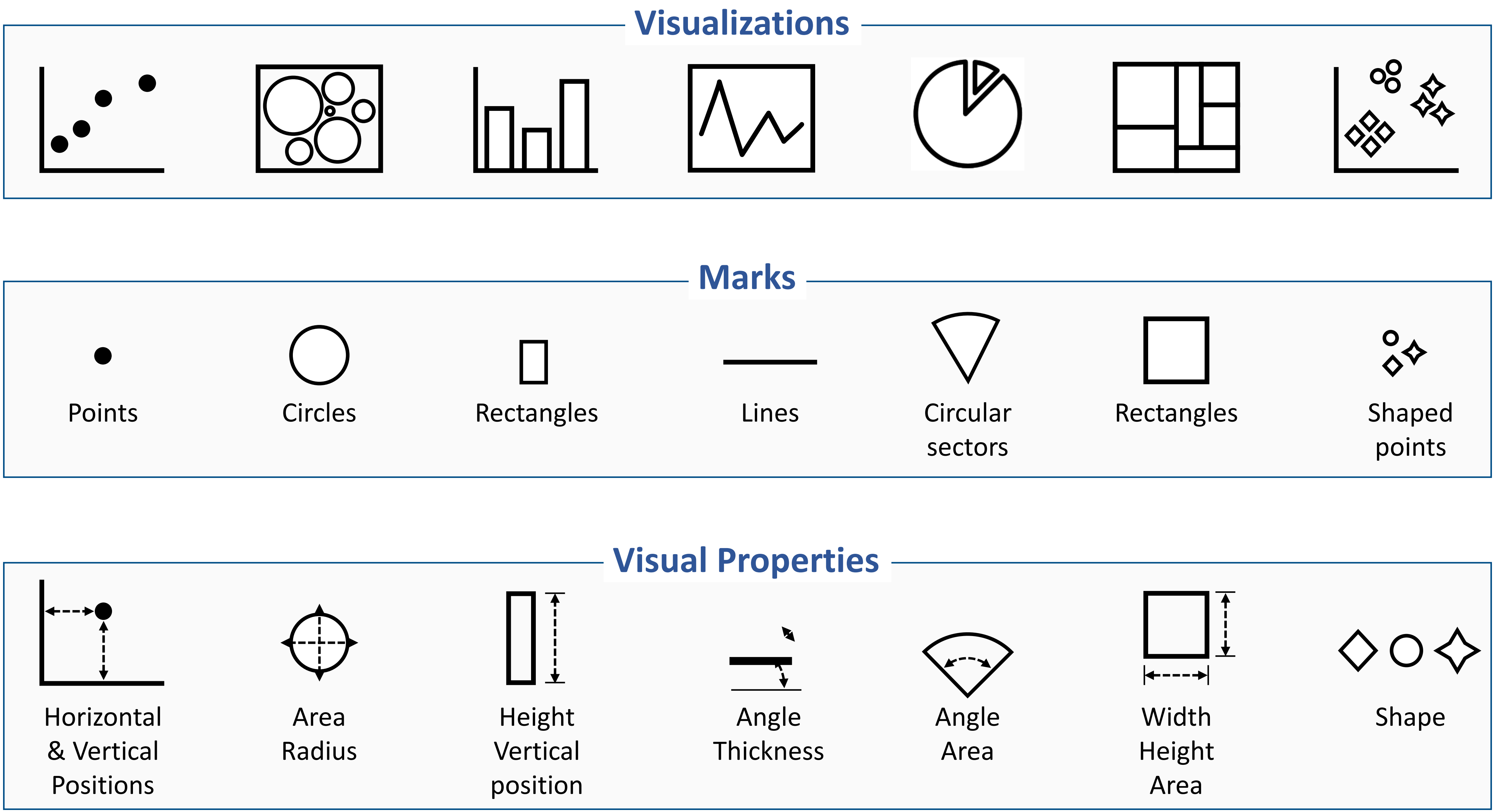
Computer-based visualization systems provide **visual representations** of **datasets** designed to help people carry out tasks more effectively.



Munzner, Tamara. Visualization Analysis and Design. A K Peters/CRC Press, 2014.

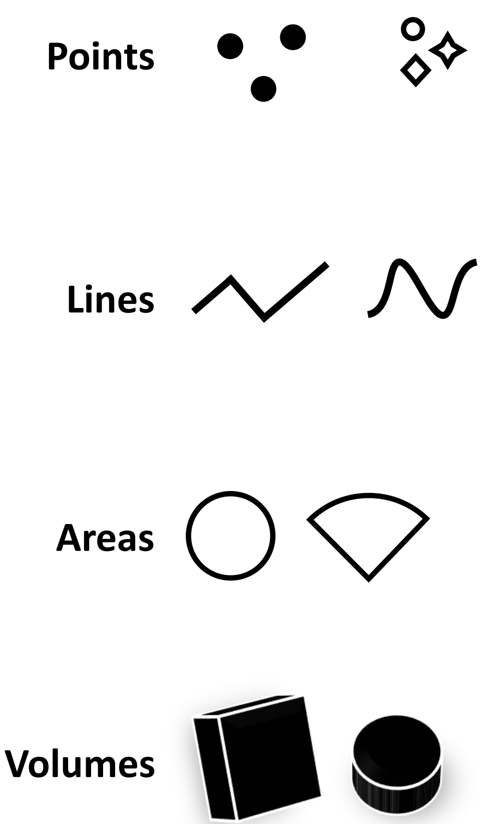


Data
Tabular
Relational
Spatial
Categorical
Ordered
Fruits
Animals
Cars
Ordinal
Quantitative
T-shirt sizes
S
M
L
–
Heights
?
tall
short
Lengths
9 inches
5 inches
–
4 inches
Legend


Visualizations
Marks
Points
Circles
Rectangles
Lines
Circular sectors
Rectangles
Shaped points
Visual Properties
Horizontal & Vertical Positions
Area Radius
Height Vertical position
Angle Thickness
Angle Area
Width Height Area
Shape

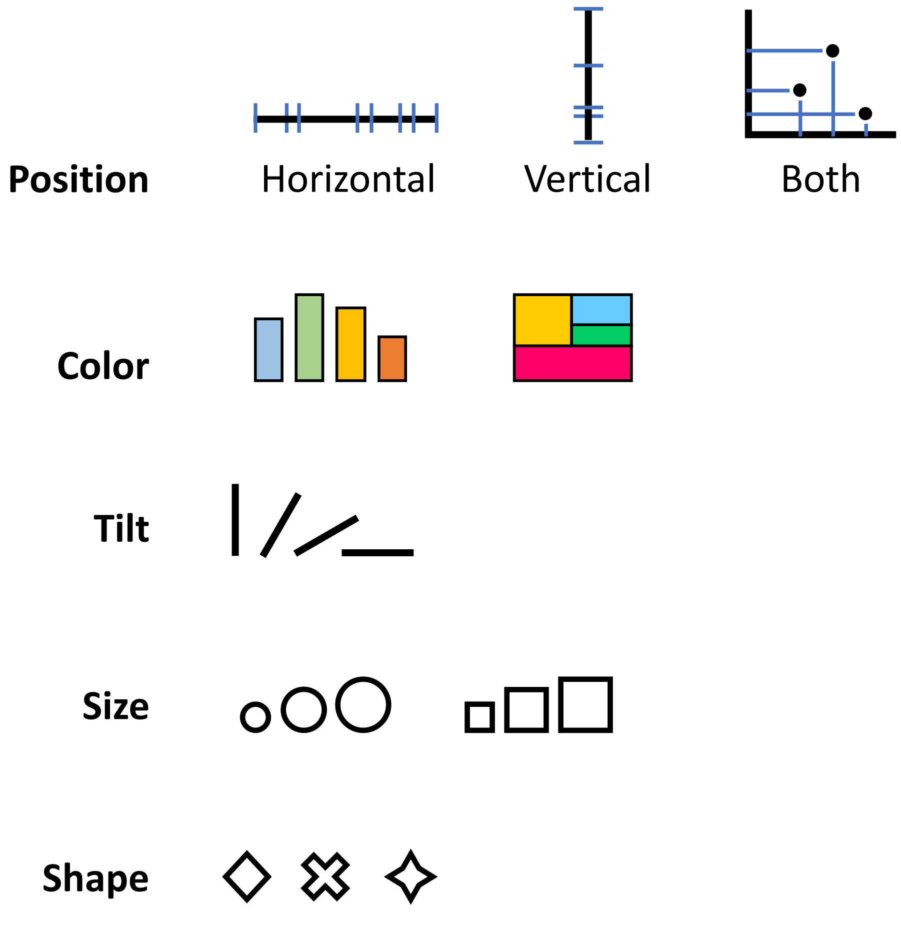
Marks

Geometric primitives that compose visualizations



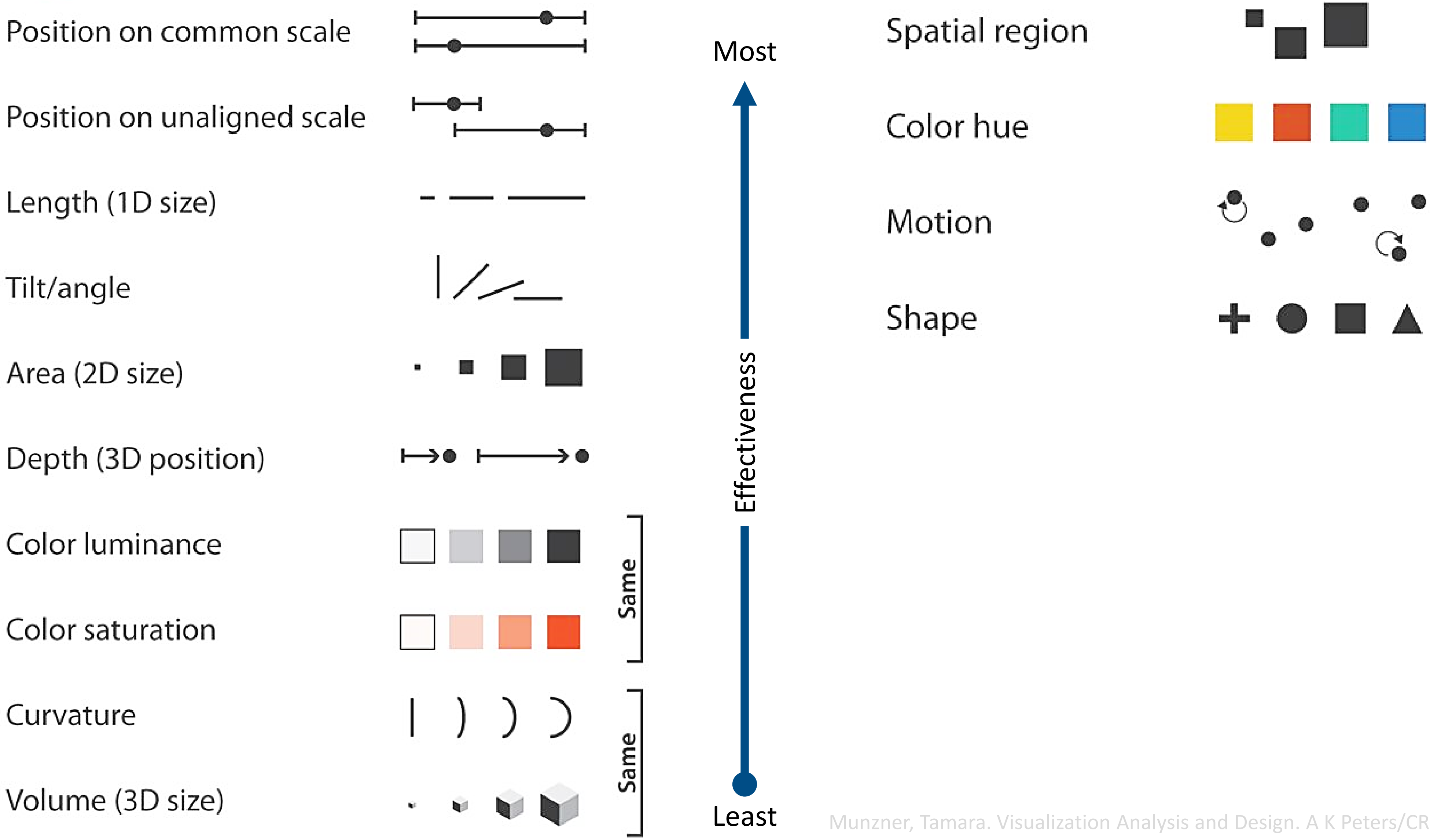
Visual Properties

Control the appearance of marks



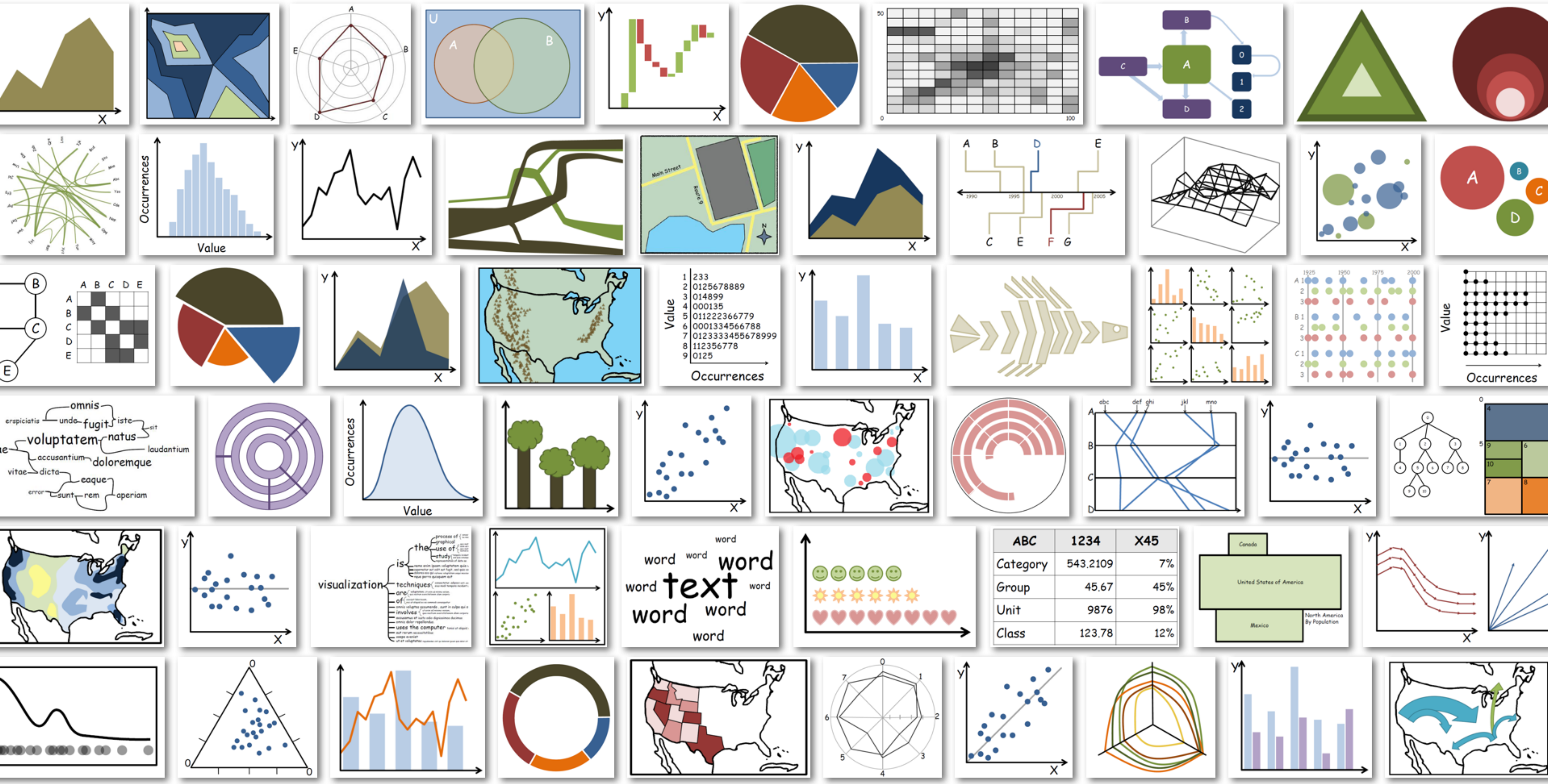
For **Ordered** data:

For **Categorical** data:



Munzner, Tamara. Visualization Analysis and Design. A K Peters/CRC Press, 2014.

Some Visualization Examples

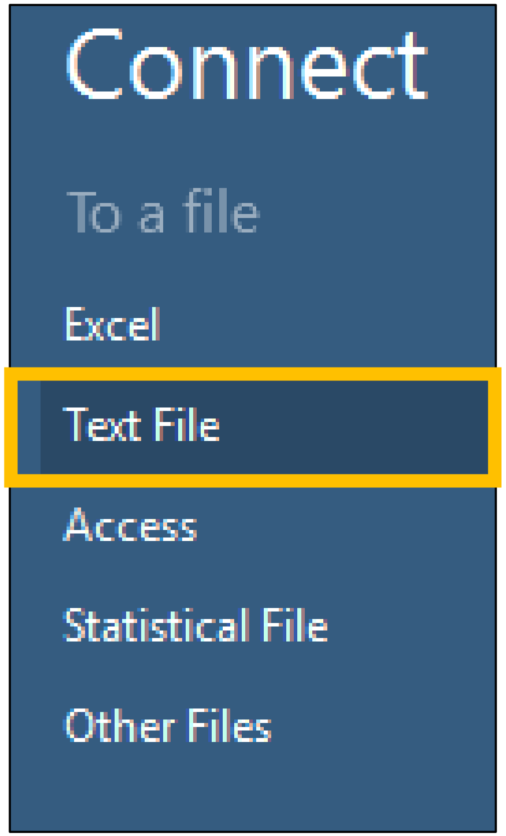


B.2 Tableau CheatSheet

Importing Data

①

Click the **Text File** option and select the **CSV dataset** with the file browser.



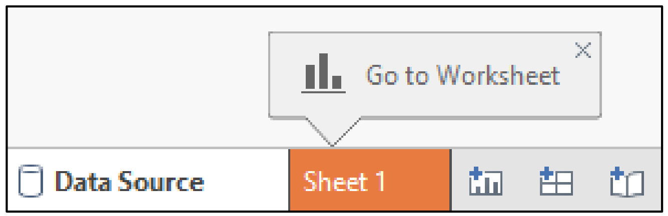
②

Your data will be shown as a table:

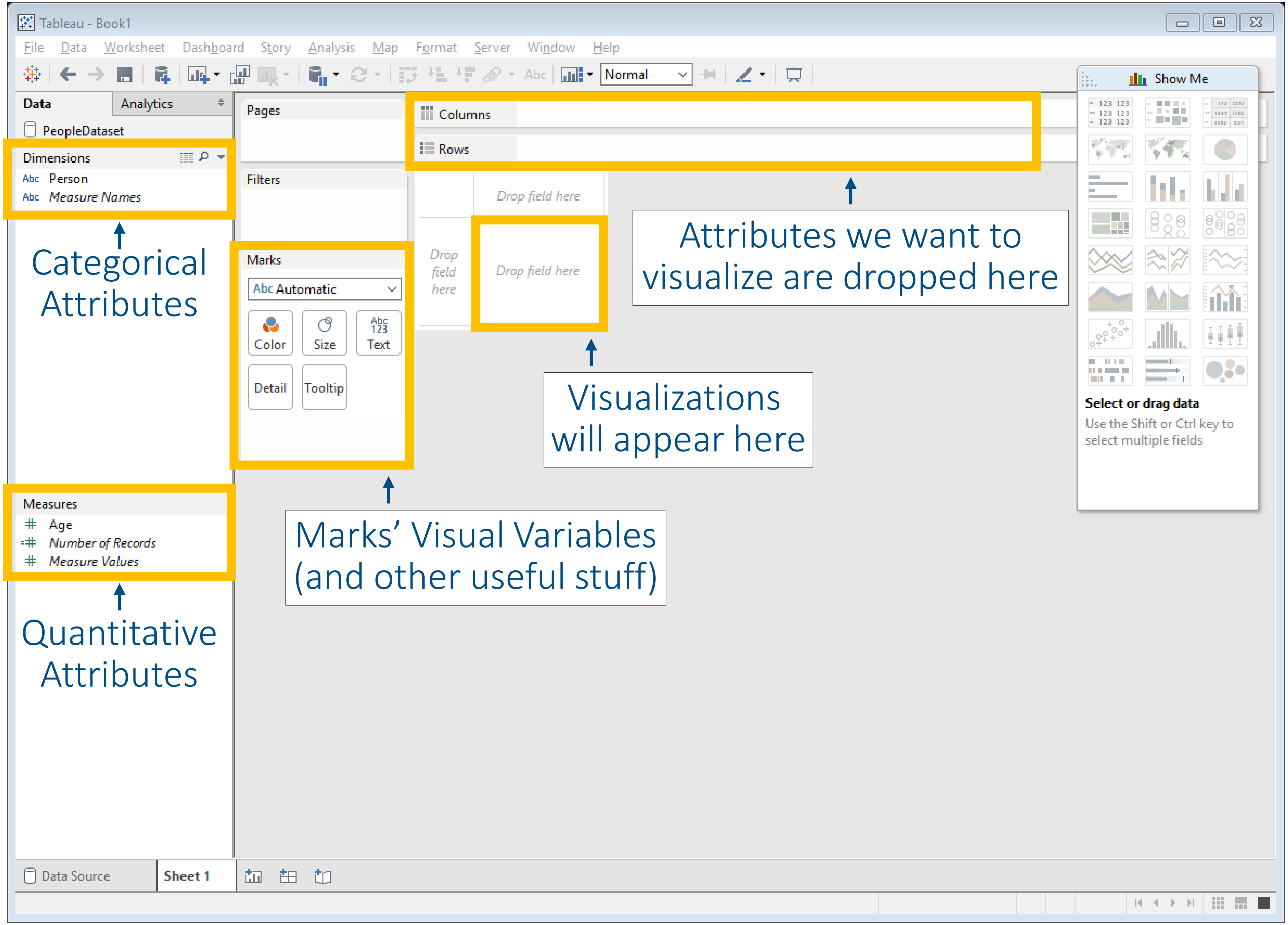
Abc PeopleDataset.csv Person	# PeopleData... Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65

③

Go to your newly created Worksheet by clicking the orange ***Sheet 1*** tab.



Worksheet Interface

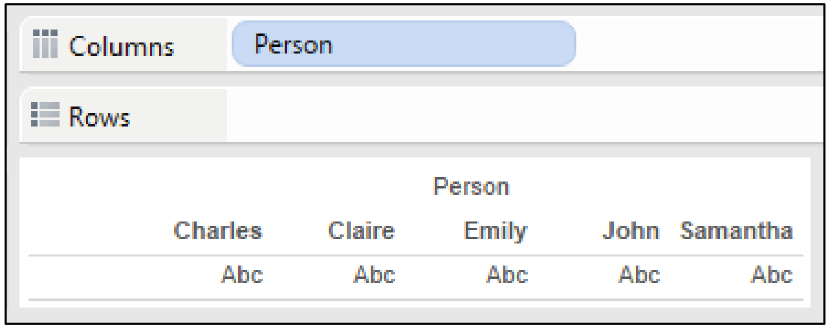


Columns and Rows Shelves

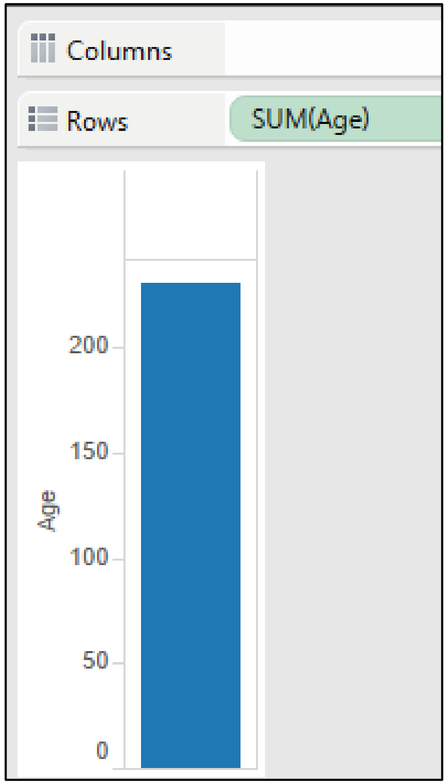
The Columns shelf creates the columns of a table, while the Rows shelf creates the rows of a table. You can place any number of fields on these shelves.

Person	Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65

Dataset



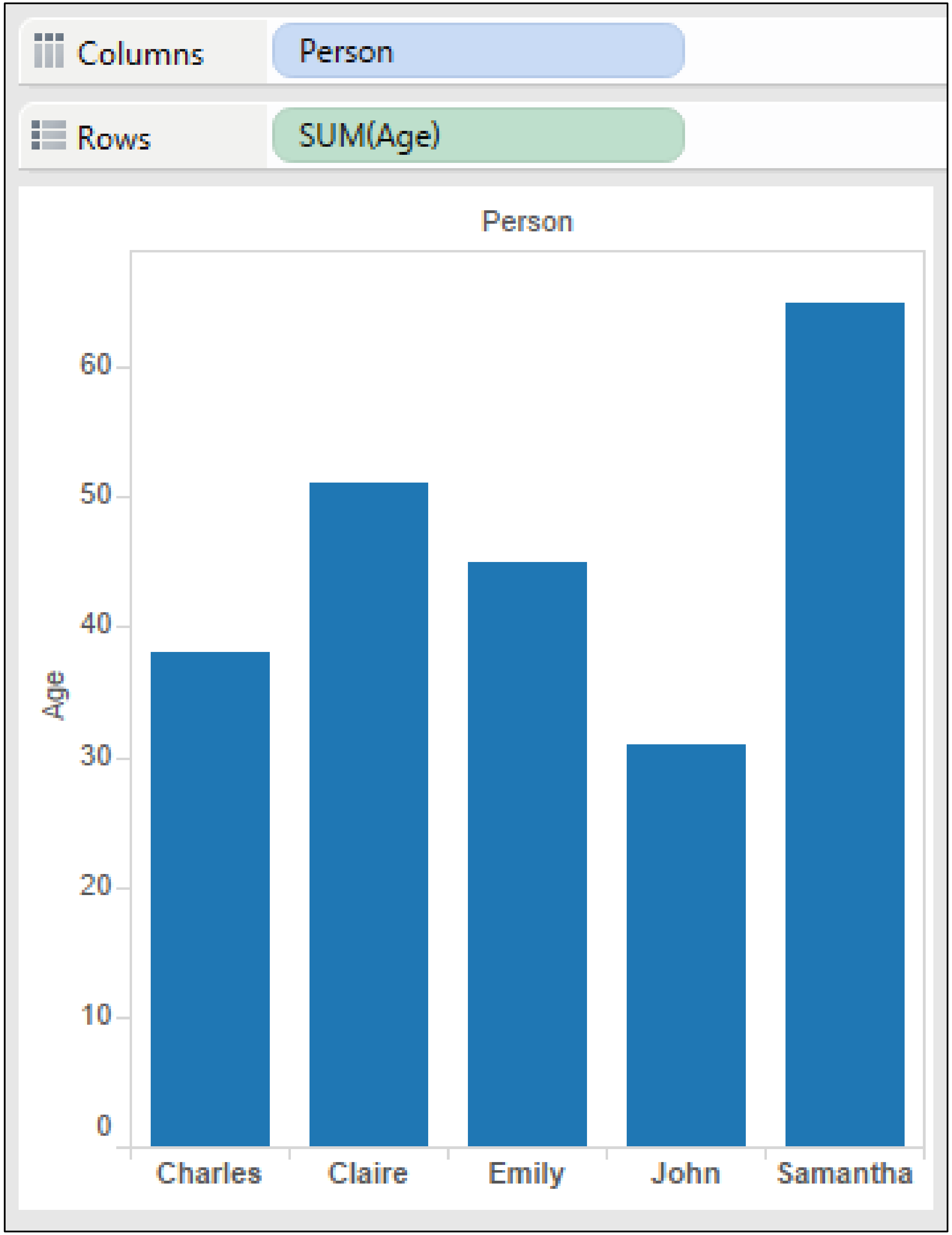
Placing a categorical attribute creates headers for the members of that category.



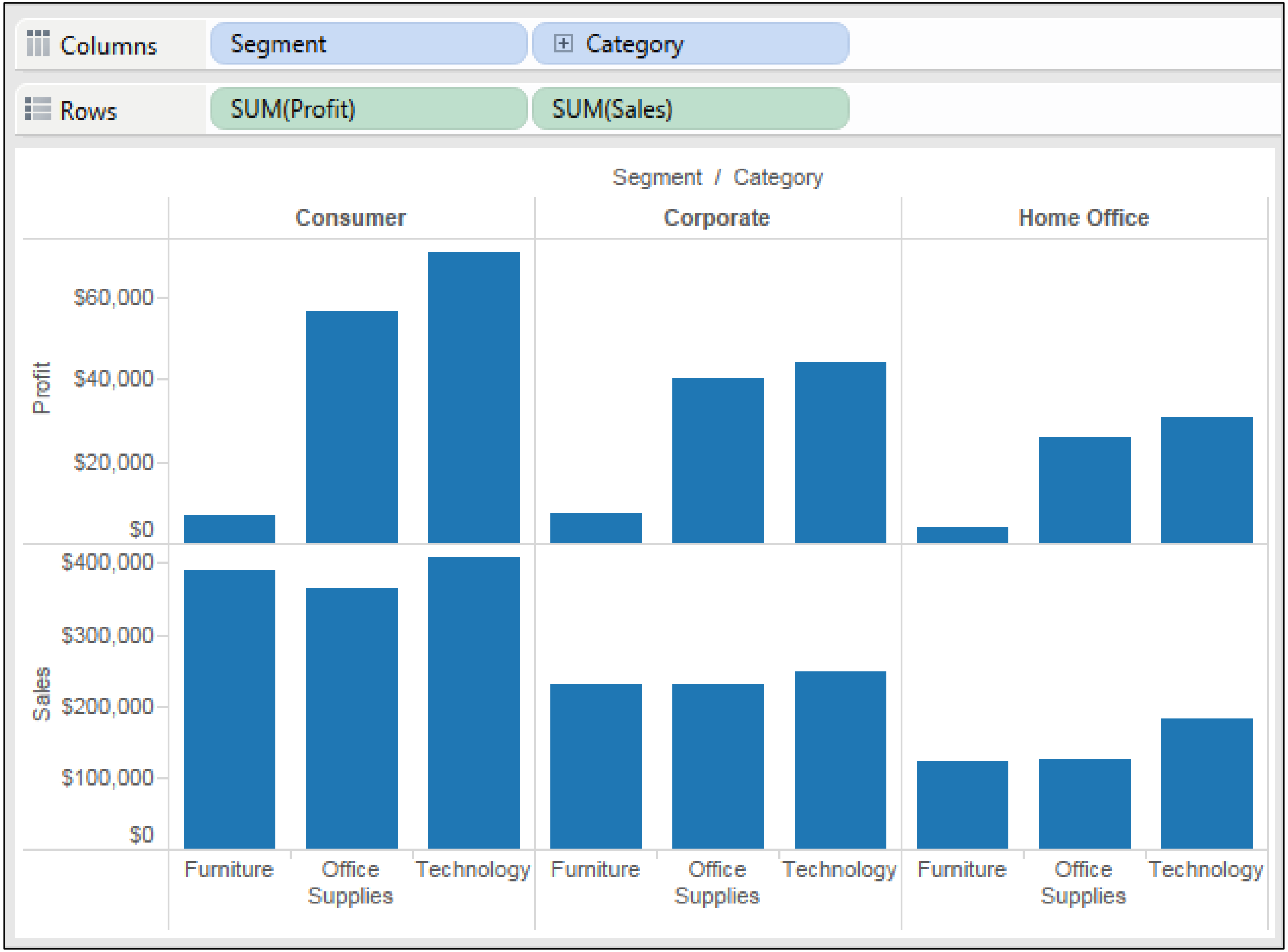
Placing quantitative data crates numeric axes

Example:

This view shows the members of the *Person* category as column headers, while the *Age* attribute is displayed as a vertical axis.

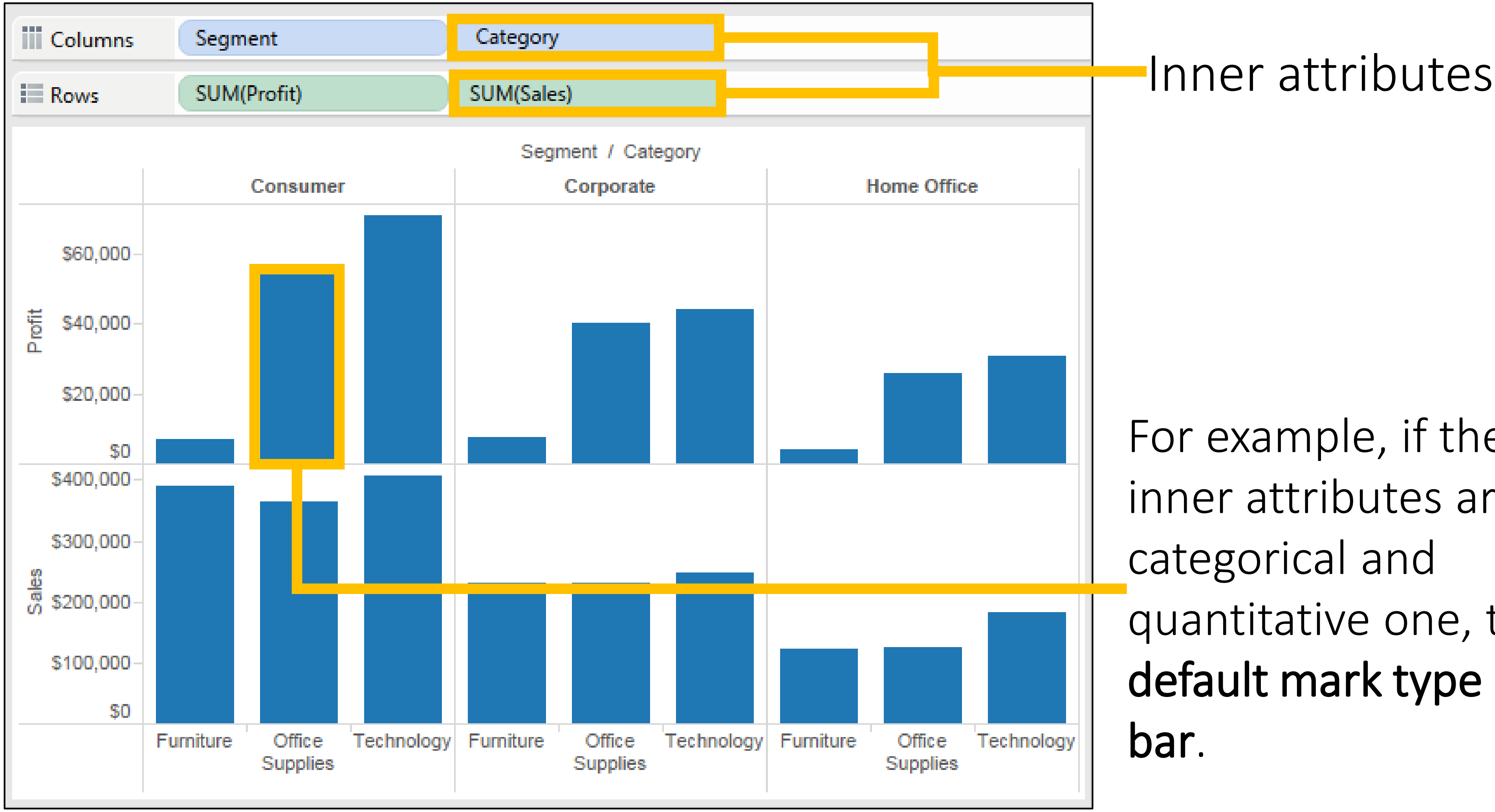


Adding more **attributes** to the ***Rows*** and ***Columns*** shelves adds more rows, columns, and panes to the table.



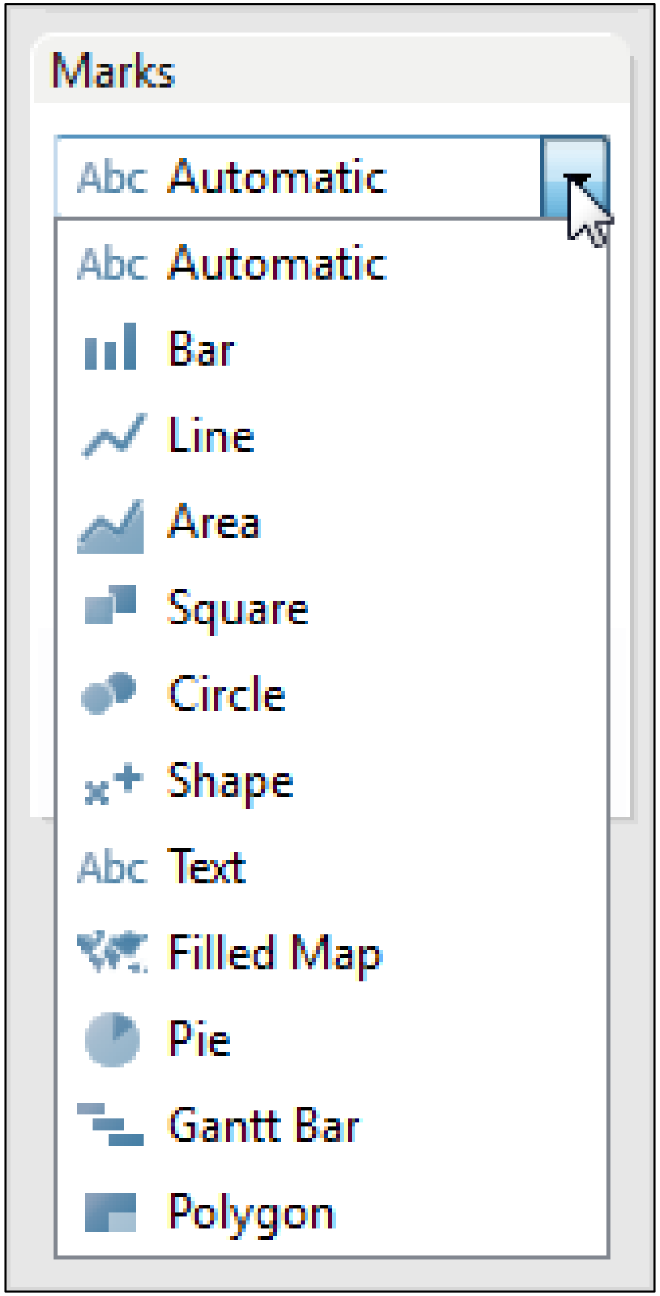
Marks

A mark is encodes the data point in the intersection of the dragged attributes. The inner attributes on the *Rows* and *Columns* shelves determine the **default mark type.**



For example, if the inner attributes are a categorical and quantitative one, the **default mark type** is a **bar**.

Marks’ Shape

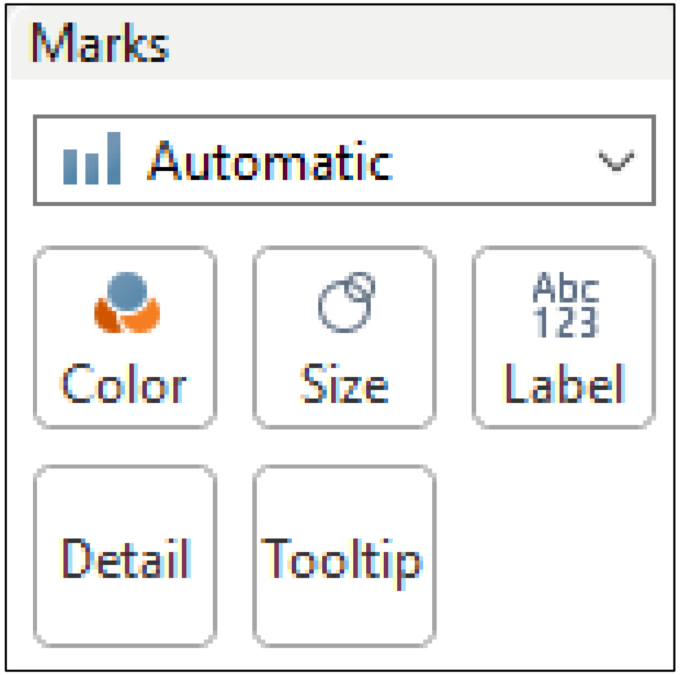


You can manually select a different mark type using the Marks card drop-down menu. This will set the mark’s **shape property**.

Other Visual Properties

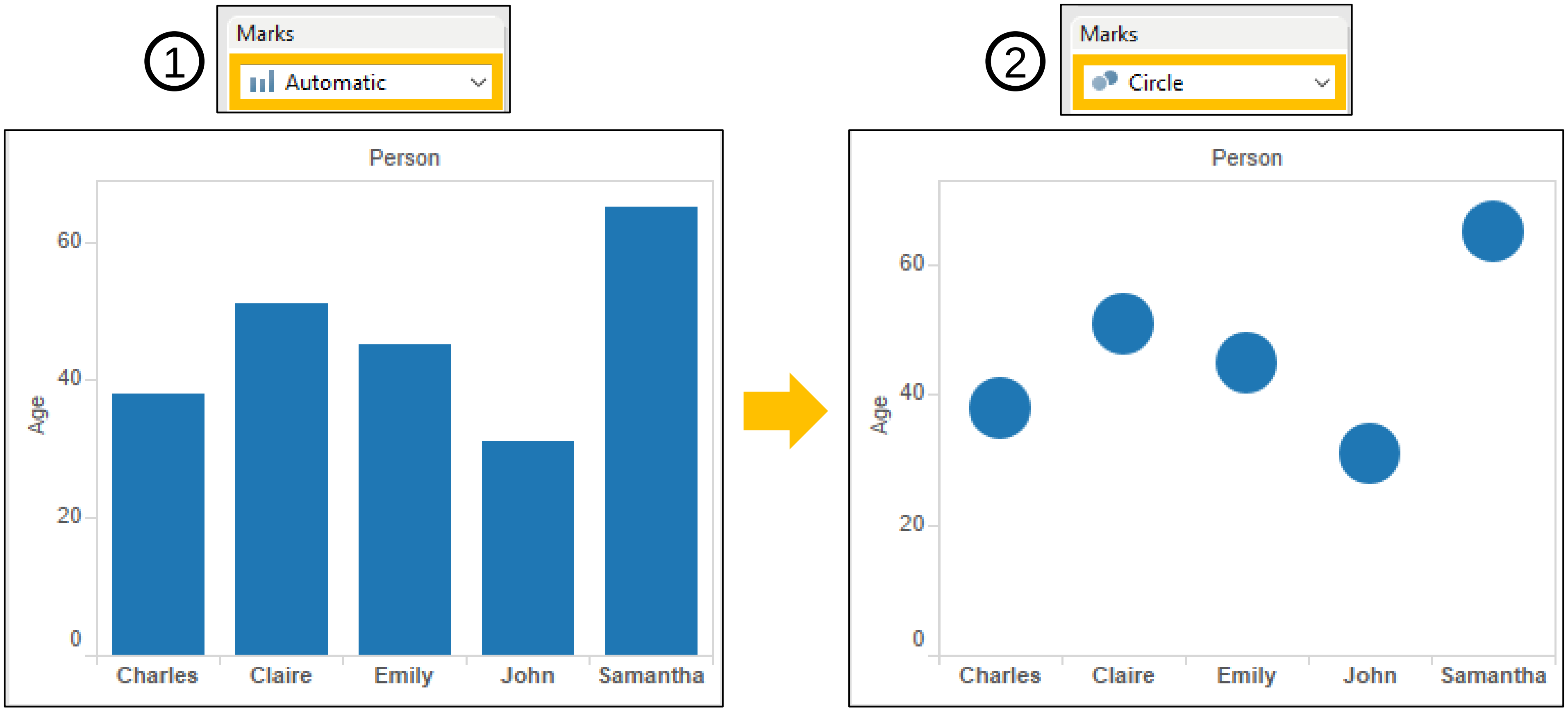
You can show additional information about the data using mark properties such as color, size, labels, etc.

Marks' properties are controlled by the ***Marks*** card. Here, you can drag attributes to the different visual properties.



Modifying the marks' shape

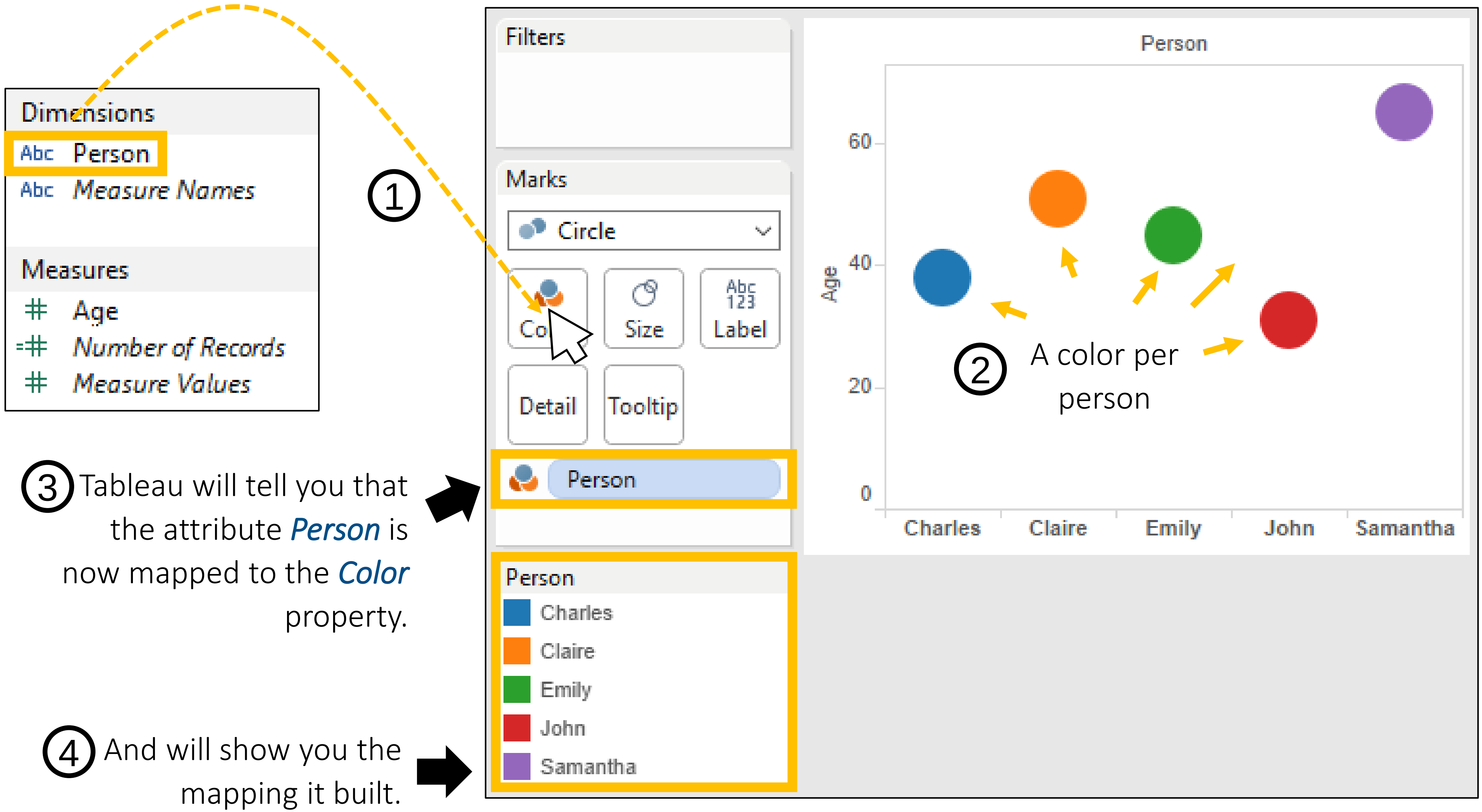
After changing the marks' type (shape) of our visualization, we end up with this:



Let's now *play* with other properties of this visualization's marks!

Coloring the marks

Dragging the *Person* attribute to the *Color* property will assign a different color for each person of the dataset.



Labelling the marks

Dragging the *Age* attribute to the *Label* property will label each mark of the visualization with the corresponding age.

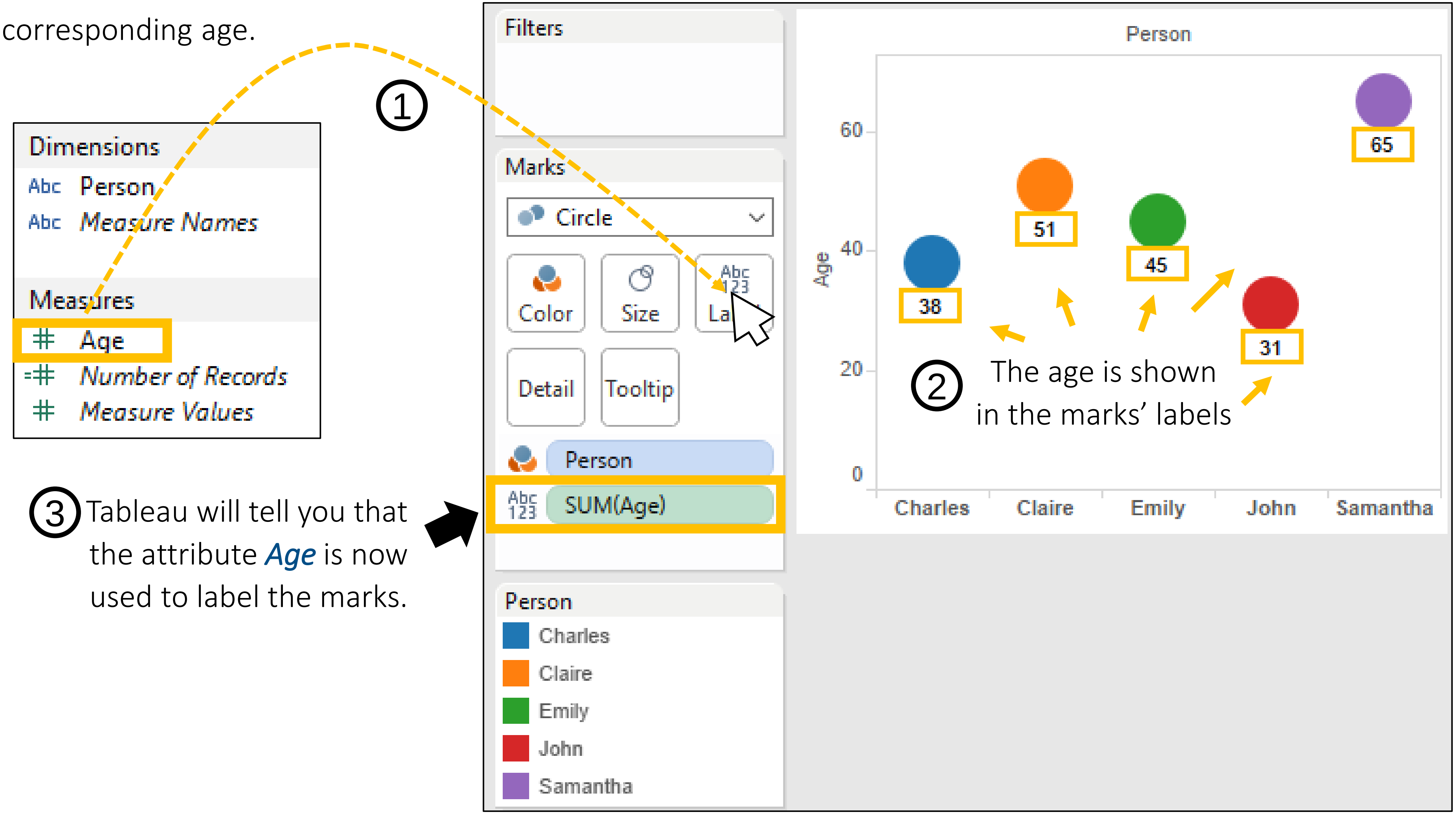


Tableau will tell you that the attribute *Age* is now used to label the marks.

The *Show me* Panel

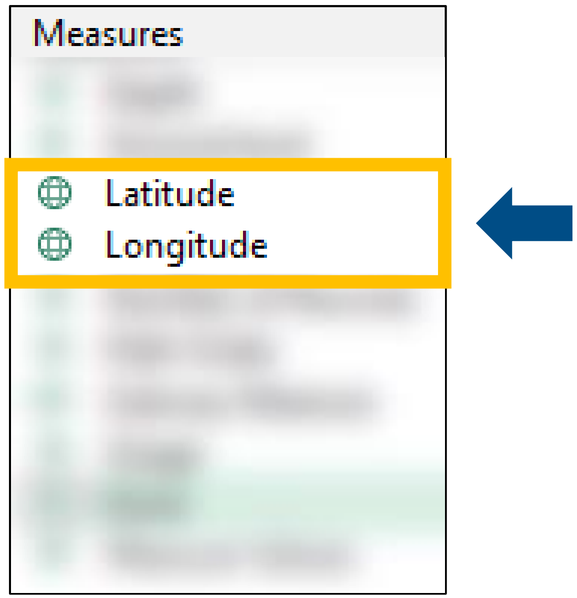
Provides suggestions to build visualizations based on the attributes you have already dropped.

Tableau automatically evaluates the selected attributes and suggests you several types of visualization that would be *"appropriate"* for those attributes.

Auto-generated Attributes

Sometimes, Tableau automatically creates attributes.

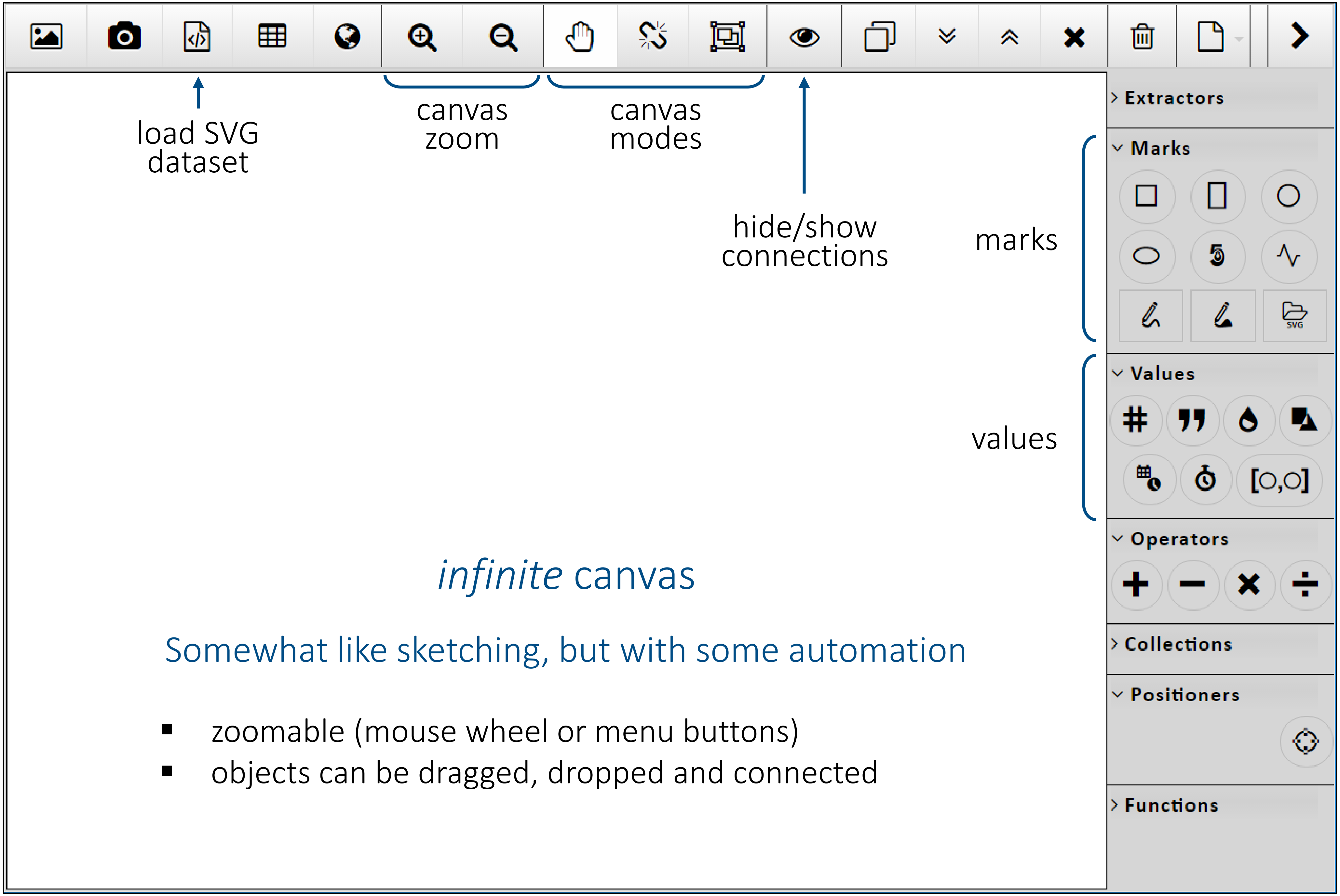
For example, when detecting geographic roles in your data (such as names of countries or cities), it associates each value in a field with a **latitude** and **longitude** values.



You can use these attributes as numeric values, for example, to place marks on top of maps.

B.3 iVoLVER CheatSheet

The Interface



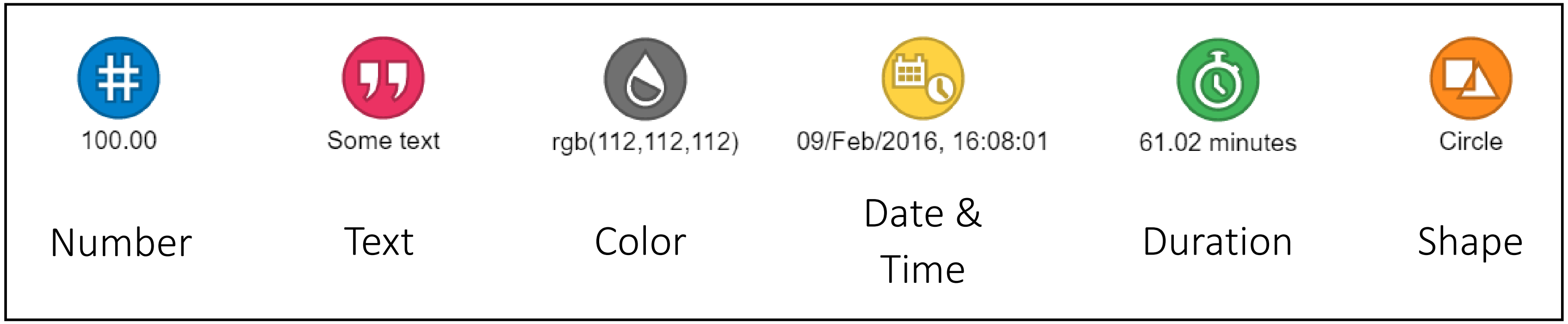
infinite canvas

Somewhat like sketching, but with some automation

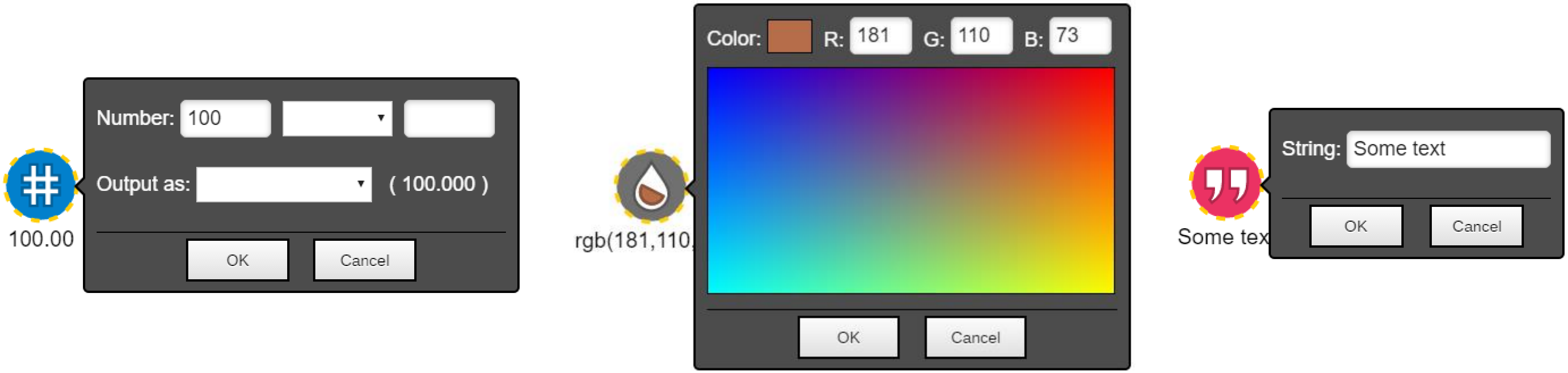
- zoomable (mouse wheel or menu buttons)
- objects can be dragged, dropped and connected

Values and Data Types

Dragged from the right toolbar. Six different types:



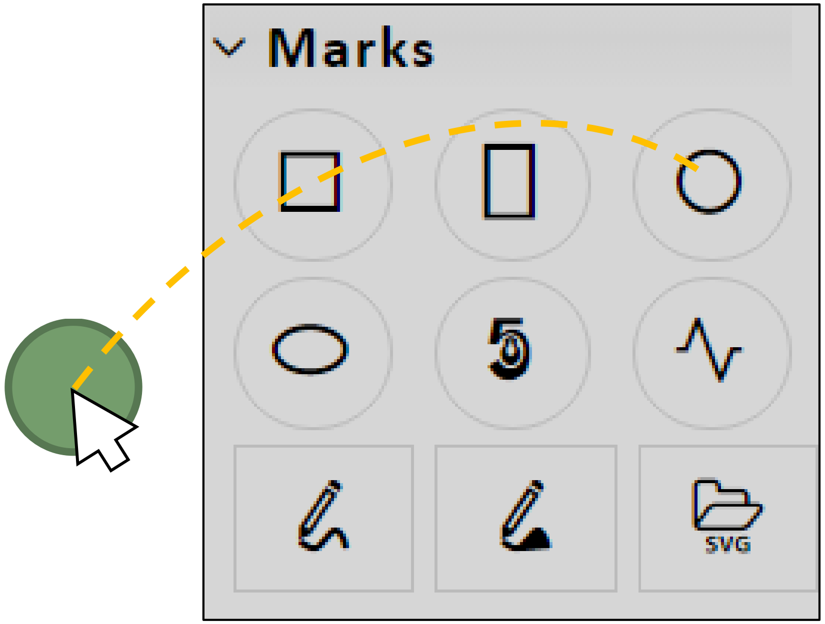
Double clicking on a value allows edition:



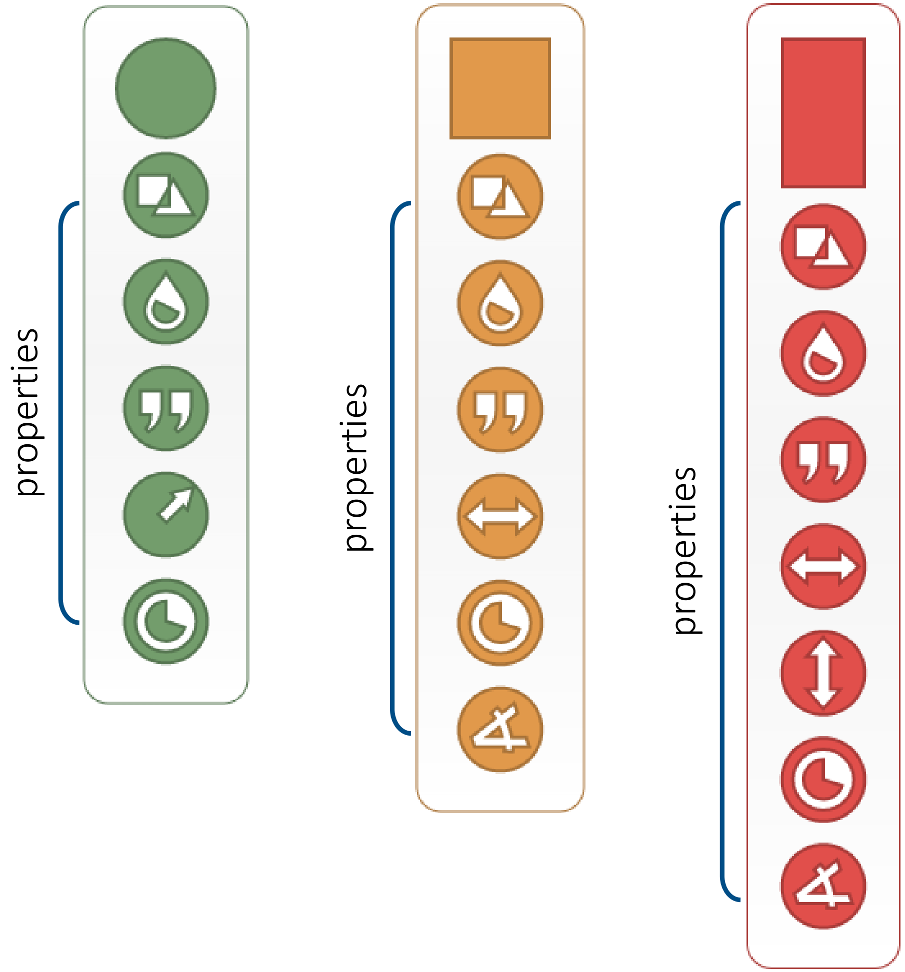
Marks

Drag them from the right toolbar and drop them on the canvas.

There are several types (squares, rectangles, circles, ellipses).



Double clicking on a mark expands it and exposes its properties. Different marks have different properties:



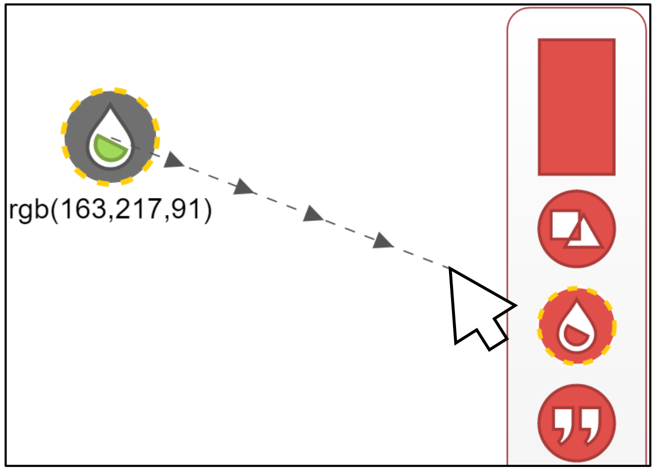
Connections

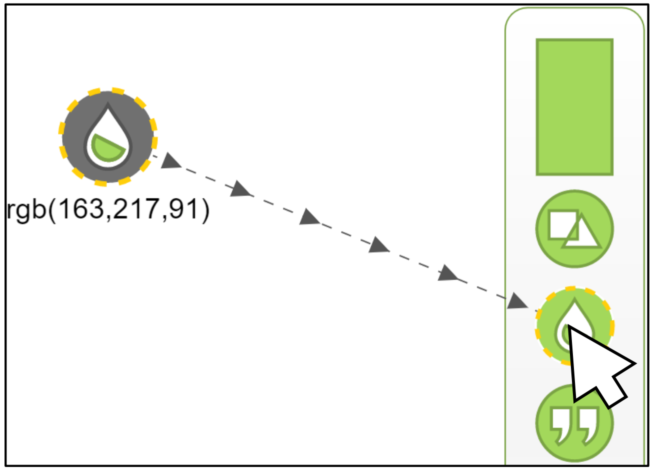
They exist **between two objects** that store a value. Indicate the **information flow** between elements.

CREATION:

1

press and hold until the value **blinks**. Then **drag** to create the connection



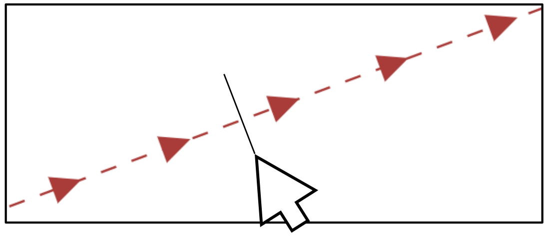


2

release the new connection **at the destination** object

DISCONNECTING:

Disconnect by **crossing** a connection while the **disconnection mode** is active.



All connections in the canvas can be **hidden** (e.g., to avoid clutter) or **shown** again with the hide/show connectors button:

SVG Datasets

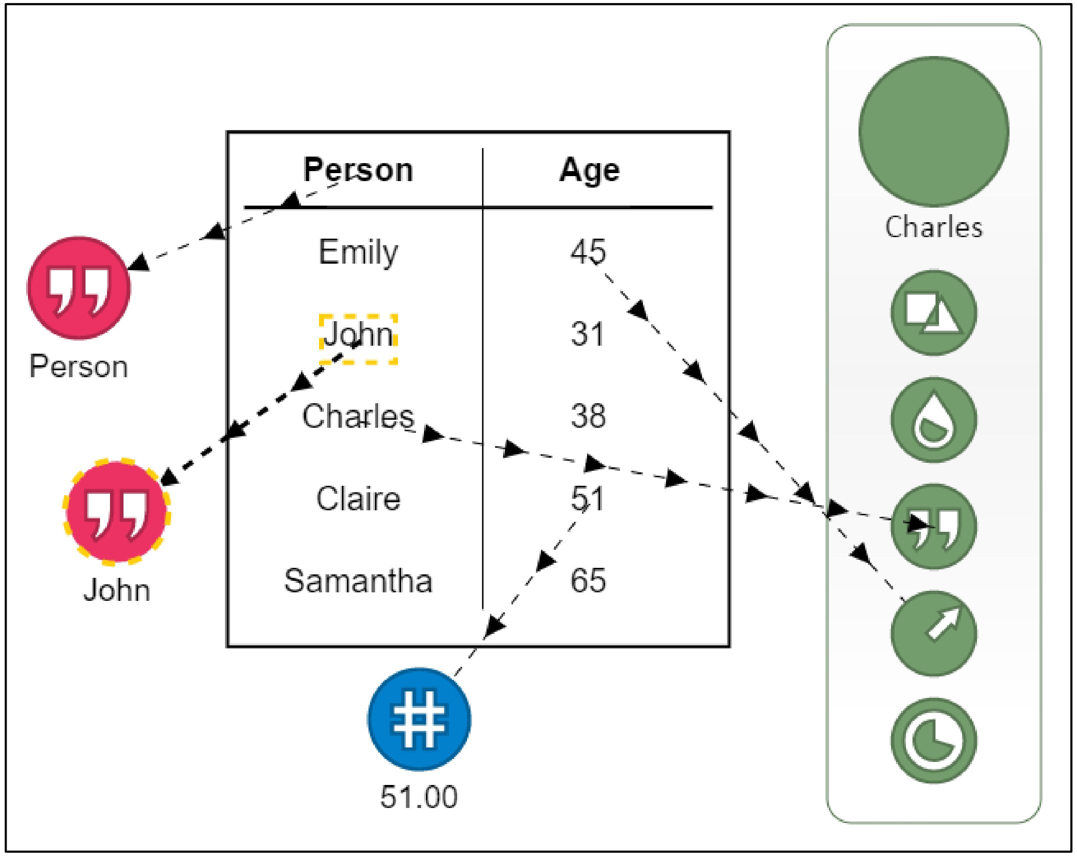
1 

Loading: Click **this icon** of the main menu and find the file you want to load.

2 The dataset will be shown in the canvas as a **draggable** table:

Person	Age
Emily	45
John	31
Charles	38
Claire	51
Samantha	65

3 The elements that compose this table are can be **output** as values or used to modify marks' properties:

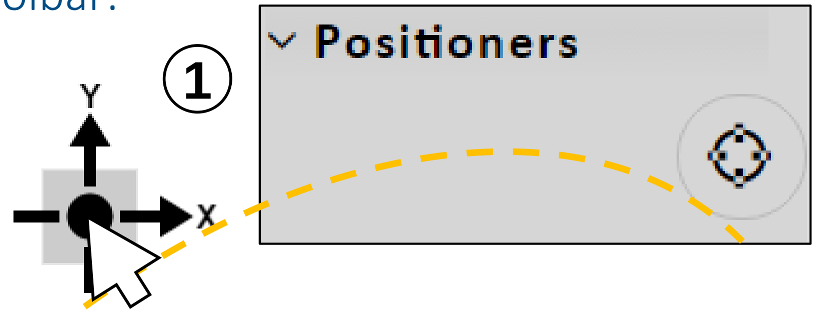


4 

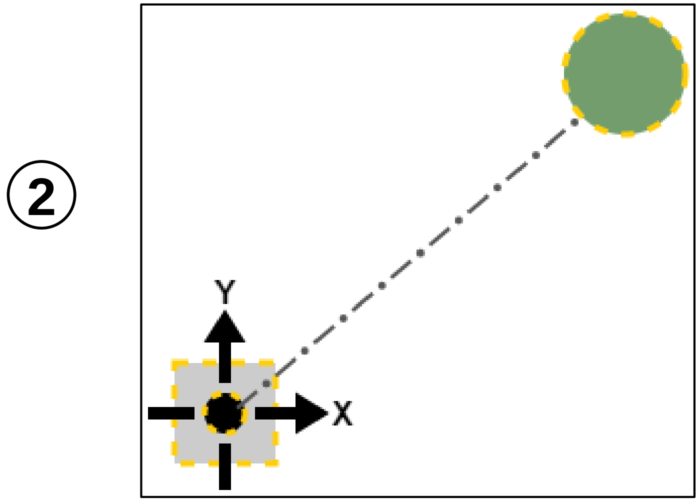
Be careful with big numbers

Positioners

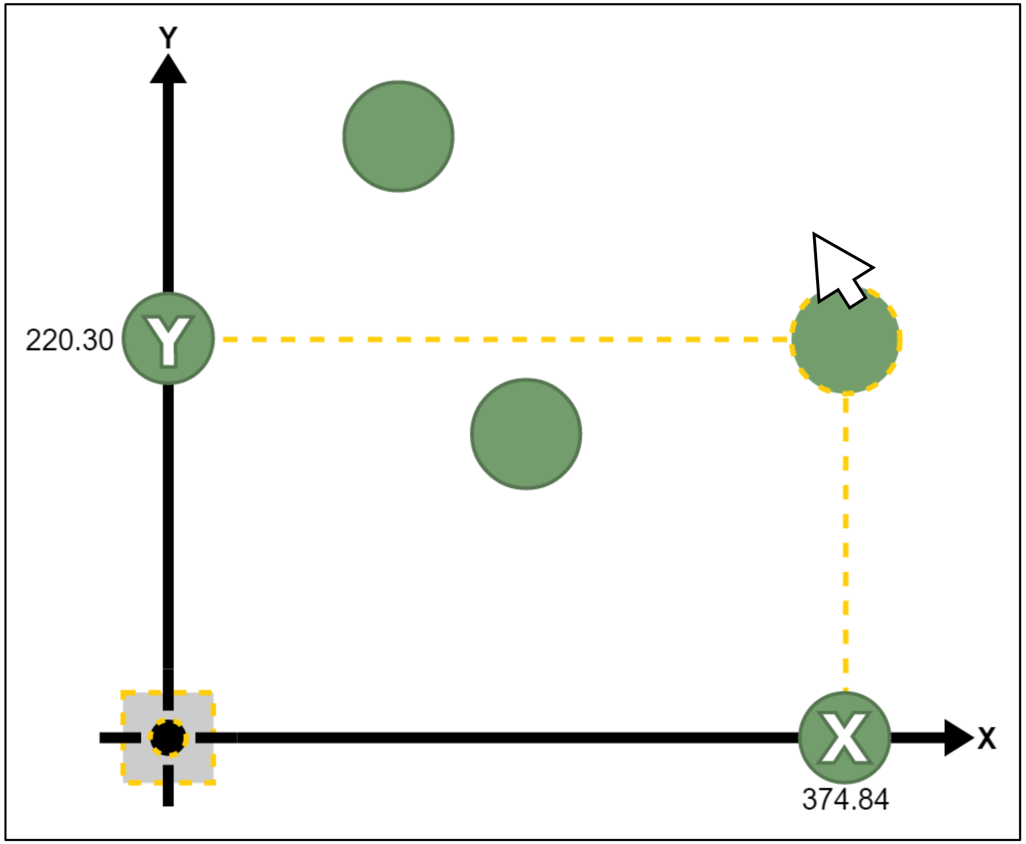
They define a **reference system** around which marks can be positioned. **Drag** them from right toolbar:



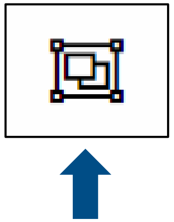
You can associated mark to a positioner by creating a link between them. **Press and hold**, wait until the positioner **blinks** and then drag the new connection to your mark:



3 **Double clicking** the positioner will expand it and you will have access to the **position properties** of its associated marks:

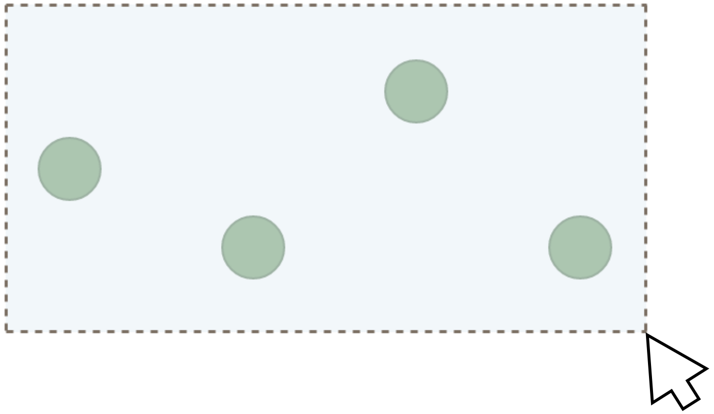


Alignment

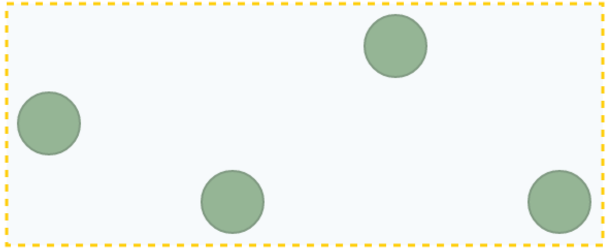
① 

Click **this icon** to activate the **mark selection** mode.

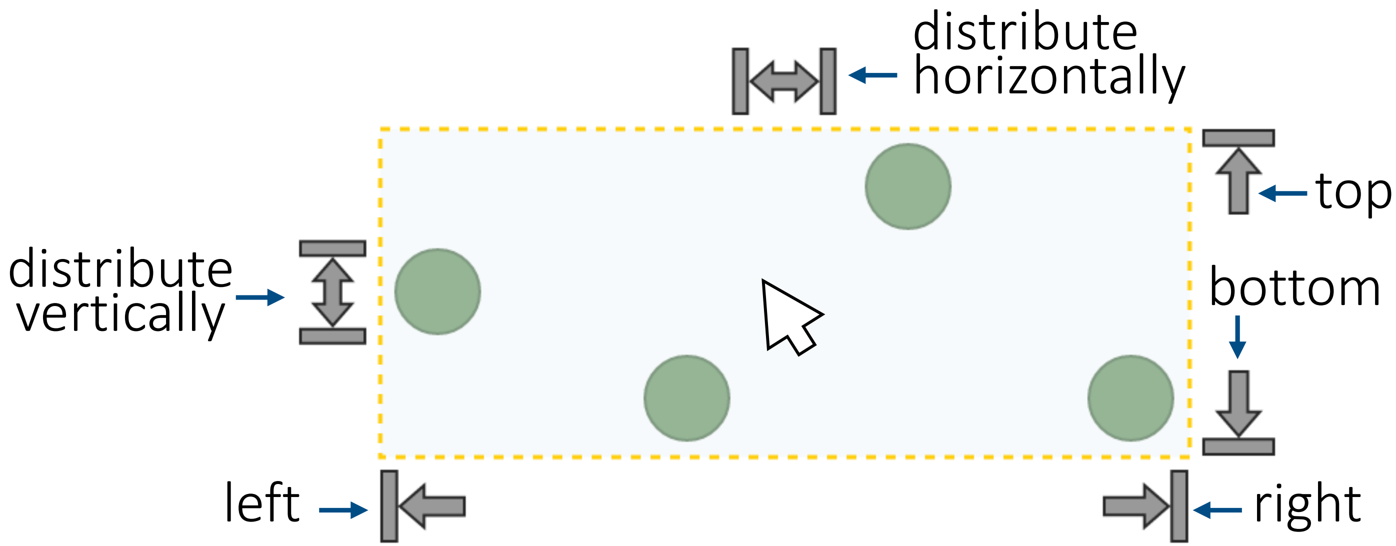
② **Click and drag** the mouse to select your **marks**.



③ Release the mouse. Your selection will appear highlighted:



④ **Double-clicking** a selection will show some **alignment options:**



⑤ **Click** an option to align the objects you selected:

Canvas and System-wide Operations

Zoom

zoom out

zoom in

Clear

Delete object

Clone mark(s)

Canvas modes

disconnection

panning

marks selection

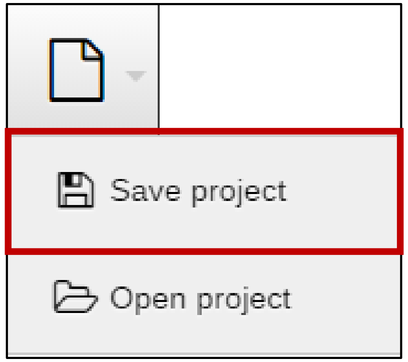
menu also available when double clicking on a blank area of the canvas

Expand/Compress All Marks

expand

compress

Saving your work



XML file

Appendix C

Experimental Material Tableau & iVoLVER Comparison: Pilot In-classroom Study

C.1 Participant Information Sheet

Participant Information Sheet

Project Title
Studying the construction of visual encodings with iVoLVER and Tableau

What is the study about?
We invite you to participate in a research project on the use of iVoLVER (the **i**nteractive **V**isual **L**anguage for **V**isualization **E**xtraction and **R**econstruction) and Tableau to implement visual encodings when learning to visualise datasets.

This study is being conducted as part of my PhD Thesis in the School of Computer Science.

Do I have to take Part?
This information sheet has been written to help you decide if you would like to take part. It is up to you and you alone whether or not to take part. If you do decide to take part you will be free to withdraw at any time without providing a reason.

Participating or not in this study or withdrawing from it will not have any effect in your learning experience in the module or in the way that you will be assessed for this module. If you decide not to take part, you will go through all the same stages as those who participate; the only difference is that we will not record or analyse your data.

What would I be required to do?
You will be given one or multiple demonstrations on two different visualization tools (iVoLVER and Tableau). After each training session, you will use these tools–in two separate lab sessions–to create visual encodings to visualise small datasets. In this process, you will apply the theory explained in the initial lectures of your CS5044 module (Information Visualisation and Visual Analytics). With your permission, we will record the screen and your interactions with the visualizations tools. In addition, at the end of the experiment, we will record you while describing and discussing your encodings in a group discussion where all the students of the module will also participate.
At the beginning of the experiment, we will ask you to provide some demographic details (e.g. age, gender) and to answer a set of multiple-choice questions (e.g. using numeric response scales) regarding your previous experience with visualization tools. After each lab session, we will ask you to fill in a questionnaire about the work that you have done and about your experience with the tool you used. We will also evaluate your learning experience and understanding of visual encodings with a quiz that will include multiple choice and open questions. We will use all the gathered data as evidence for discoveries in our research publications and presentations.

Will my participation be Anonymous and Confidential?
Only the researchers and the supervisor will have access to the raw data which will be kept strictly confidential. We will take notes, which will be anonymous. The results from the questionnaires will not be associated to your name in any publications; only the researchers will be able to connect your comments and answers to the questionnaires to your identity. If the videos and photos that we will take show your face, your voice, or identifiable parts of your body or your behaviour, we will generate anonymized versions of these materials (e.g. blurring your face and transcribing your spoken statements). We ask for explicit permission from you to share these anonymized versions of the video snippets or pictures in our publications. Please read carefully and answer the questions in the Consent form that accompanies this information sheet.

Storage and Destruction of Data Collected

The raw data collected during the study will be accessible by the researchers and the supervisor involved in this study only and stored for a maximum of 5 years. During its existence, raw data will be held securely in an office in the University of St Andrews' Department of Computer Science and in the securely encrypted data drives of the researchers.

What will happen to the results of the research study?

The results will be finalised by approximately 2018 and written up as part of my PhD Thesis and possibly as other research publications and presentations.

Are there any potential risks to taking part?

The risks of taking part should be no different from those you experience on a normal lecture or lab session.

Questions

You will have the opportunity to ask any questions in relation to this project before completing the Consent Form attached.

Consent and Approval

This research proposal has been scrutinised and been granted Ethical Approval through the University ethical approval process.

What should I do if I have concerns about this study?

A full outline of the procedures governed by the University Teaching and Research Ethical Committee is available at http://www.st-andrews.ac.uk/utrec/Guidelines/complaints
Contact details of the researchers are below.

Contact Details

Researcher: Gonzalo Gabriel Méndez
Contact Details: ggm@st-andrews.ac.uk

Researcher: Uta Hinrichs
Contact Details: uh3@st-andrews.ac.uk

Supervisor: Dr. Miguel Nacenta
Contact Details: mans@st-andrews.ac.uk Phone: +44 (0)1334 46 3265

C.2 Participant Consent Form – Coded Data

Participant Consent Form
Coded Data

Project Title
Studying the construction of visual encodings with iVoLVER and Tableau

Researcher(s) Name(s)

Gonzalo Gabriel Méndez
School of Computer Science
University of St Andrews
ggm@st-andrews.ac.uk

Dr. Uta Hinrichs
School of Computer Science
University of St Andrews
uh3@st-andrews.ac.uk

Supervisors Names

Dr. Miguel Nacenta
School of Computer Science
University of St Andrews
mans@st-andrews.ac.uk

The University of St Andrews attaches high priority to the ethical conduct of research. We therefore ask you to consider the following points before signing this form. Your signature confirms that you are happy to participate in the study.

What is Coded Data?
The term ‘Coded Data’ refers to when data collected by the researcher is identifiable as belonging to a particular participant but is kept with personal identifiers removed. The researcher(s) retain a ‘key’ to the coded data which allows individual participants to be re-connected with their data at a later date if necessary. The un-coded data is kept confidential to the researcher(s) (and Supervisors). If consent is given to archive data (see consent section of form) the participant may be contacted in the future by the original researcher(s) or other researcher(s).

Consent
The purpose of this form is to ensure that you are willing to take part in this study and to let you understand what it entails. Signing this form does not commit you to anything you do not wish to do and you are free to withdraw at any stage and without providing a reason.
Material gathered during this research will be coded and kept confidentially by the researcher, with only the researchers and supervisor listed above having access. All data will be securely stored in an office at the Computer Science Department of the University of St Andrews for a maximum of 5 years. Video snippets, photographs taken during the study sessions and data/statements that you provide via the questionnaires or during the group discussion may be used as part of research publications and presentations in anonymised form.
Please answer each statement concerning the collection and use of the research data.

I have read and understood the information sheet.	☐ Yes	☐ No
I have been given the opportunity to ask questions about the study.	☐ Yes	☐ No
I have had my questions answered satisfactorily.	☐ Yes	☐ No
I understand that I can withdraw from the study at any time without having to give an explanation.	☐ Yes	☐ No
I understand that participating, not participating, or withdrawing from the study at any point will not have any effect on the evaluation or learning process of the CS5044 module.	☐ Yes	☐ No
I understand that my data will contain identifiable personal data, but that will be stored with personal identifiers removed by the researcher and that only the researchers and the supervisor listed above will be able to decode this information as and when necessary.	☐ Yes	☐ No
I understand that my data will be stored for a period of up to 5 years before being destroyed	☐ Yes	☐ No
I have been made fully aware of the potential risks associated with this research, and I am satisfied with the information provided.	☐ Yes	☐ No
I agree to take part in the study.	☐ Yes	☐ No

Part of our research involves taking video- and audio recordings as well as photographs. These recordings will be kept secure and stored with no identifying documents i.e. consent forms and questionnaires. With your consent, we would like to use this data in anonymised form as part of publications and presentations. We will blur any identifying parts in the video or photographs and transcribe verbal statements so that you cannot be identified based on your voice. However, you should know that there is still a chance that you are identified in these photographs and video recordings.

Photographs and video/audio are highly valuable resources for future studies. We therefore ask for your additional consent to maintain data and images for this purpose.

I agree to be video- and audio recorded and photographed during the study sessions.	☐ Yes	☐ No
I agree for my video, audio and photographic recorded material to be presented and published in anonymised form as part of this research. I understand that even in anonymised form video, audio and photos may still allow others to identify me.	☐ Yes	☐ No
I agree for all recorded data to be used in future studies.	☐ Yes	☐ No

Participation in this research is completely voluntary and your consent is required before you can participate in this research. If you decide at a later date that data should be destroyed we will honour your request in writing.

Name in Block Capitals ______________________________

Signature ______________________________

Date ______________________________

C.3 Demographics and Experience Questionnaire

Study Constructing Visual Encodings

*Required

Demographic Information

1. **What is your age? ***

2. **What is your gender? ***
Mark only one oval.
- Female
- Male
- Other

3. **What is your current level of study? ***
Mark only one oval.
- Undergraduate
- Graduate

4. **What is your area of study? ***
e.g., Middle Eastern Literary and Cultural Studies, Art History and Mathematics, Computer Science

About your visualization experience

In the following questions, select the option that best represents your answer regarding your personal experience and background with visualizations:

5. **How familiar are you with Information Visualization? ***
Select the option that applies best:
Mark only one oval.
- 1. I do not know anything about information visualization at all.
- 2. I just started to explore visualization topics as part of the CS5044 module. Before, I did not have any knowledge about it at all.
- 3. I have come across information visualizations on the web and in magazines, but before this module I never considered the concepts and theory behind them.
- 4. I have been actively following and reading about visualization related topics, but I do not have much experience with practically applying visualization concepts myself.
- 5. I know what information visualization is; I am actively reading about the topic and I have been creating visualizations myself for at least two years.

6. **How familiar are you with reading and creating graphs and charts?** *

E.g. bar charts, line graphs

Mark only one oval.

- 1. I do not really know what graphs and charts are.
- 2. I have seen charts in newspapers and the like but I have not paid much attention to them.
- 3. I know graphs and charts from the newspapers and the web and I know how to read them.
- 4. I know how to read graphs and charts and I have created some on my own a couple of times.
- 5. I feel very comfortable at both reading and creating graphs and charts and I do it often.

7. **How often do you create information visualizations?** *

Mark only one oval.

- 1. Never.
- 2. I have sketched out abstract concepts and/or data representations on paper before, but never created a visualization using computational tools.
- 3. I have created some static visualizations (with Excel, for example).
- 4. I regularly create basic charts and I have created at least one interactive visualization.
- 5. I create interactive visualizations regularly as part of my studies/research/job.

8. **How would you rate your experience with the following visualization tools:** *

Mark only one oval per row.

	I do not know this tool	I am aware of it, but I've never used it	I have used it once	I am comfortable with it, but not proficient	I consider myself proficient
Tableau Desktop					
MS Excel					
D3.js					
Processing					
Google Fusion Tables					
Gephi					
Python / Bokeh / iPython Notebook					
Matlab					
R/Shiny					

C.4 Tool Questionnaire

Study Constructing Visual Encodings

*Required

Tool Questionnaire

The following questions refer to the tool you used during this practical.

1. **What tool did you use today? ***
   Select the tool you used in the practical of today
   Mark only one oval.

   - iVoLVER
   - Tableau

2. **How satisfied are you with your overall experience using this tool to create visualizations? ***
   Mark only one oval.

	1	2	3	4	5	
Very Dissatisfied						Very Satisfied

3. **What best describes the process of constructing viusualizations with this tool? ***
   Mark only one oval.

	1	2	3	4	5	
Very Difficult						Very Easy

4. **In your opinion, how well does this tool reflect what you have learned so far about constructing visualizations? ***
   Mark only one oval.

	1	2	3	4	5	
Very well						Very badly

5. *
   Explain why

6. **Which features of this tool did you find most useful or did you like the most?**

7. **Which features of this tool did you find less useful or encounter problems with?**
   Describe why

8. **How would you improve this tool?**

C.5 Visualizations Questionnaire

Study Constructing Visual Encodings

*Required

Section 1: About your housing dataset visualization sketch

The following questions refer to the visualization of the housing dataset you worked on in the second part of the tutorial (Exercise 2). Please, answer them considering the final version of your sketch. If you have created several sketches, consider the one that you think is the most effective.

1. **How satisfied are you with your visualization sketch?** *
   Mark only one oval.

	1	2	3	4	5	
Very disatisfied	◯	◯	◯	◯	◯	Very satisfied

2. **What aspects of the data does your visualization sketch highlight and how?** *

3. **How did you decide on the visual encoding of data in your visualization sketch?** *

4. **Did you consider a different visual encoding while working with the software? If so, describe the different designs you considered and explain why you discarded them at the end?** *

5. **With more time, what aspects of your visualization would you change? ***
   Provide a brief reason why you would apply them.

6. **Describe, step by step, how Alice and Bob (the people mentioned in the scenario) would use your visualization to find candidate houses to buy.**

Section 2: About the housing dataset

7. **Please specify the data attributes that your visualization shows as well as their type.**
   Mark only one oval per row.

	Categorical	Ordinal	Quantitative
Year Built	◯	◯	◯
Garage	◯	◯	◯
Price	◯	◯	◯
Footage Class	◯	◯	◯
House ID	◯	◯	◯
Distance to Shops	◯	◯	◯
Type	◯	◯	◯
# of Bedrooms	◯	◯	◯
Lot Width	◯	◯	◯

8. **Which visual variables did you use to encode the following attributes of the dataset?**
   # of Bedrooms:

9. Type:

10. Lot Width:

11. Price:

12. Year Built:

13. Garage:

14. Distance to Town:

15. Footage Class:

16. **How do you make use of hue or value in your visualization (if at all)?**
Provide a short explanation on how/what for you are using colors in your sketch

17. **How do you make use of position in your visualization (if at all)?**
Provide a short explanation on how/what for you are using positions in your sketch

Appendix D

Experimental Material Tableau & iVoLVER Comparison: Main Lab Controlled Study

D.1 Participant Information Sheet

Participant Information Sheet

University of St Andrews

Project Title
Studying the construction of visual encodings with iVoLVER and Tableau

What is the study about?
We invite you to participate in a research project on the use of iVoLVER (the **i**nteractive **V**isual **L**anguage for **V**isualization **E**xtraction and **R**econstruction) and Tableau to implement visual encodings when learning to visualise datasets.

This study is being conducted as part of my PhD Thesis in the School of Computer Science.

Do I have to take Part?
This information sheet has been written to help you decide if you would like to take part. It is up to you and you alone whether or not to take part. If you do decide to take part you will be free to withdraw at any time without providing a reason.

What would I be required to do?
At the beginning of the experiment, we will ask you to provide some demographic details (e.g. age, gender) and to answer a set of multiple-choice questions (e.g. using numeric response scales) regarding your previous experience with visualization tools. You will then be asked to use two different visualization tools to build visualizations of a given data set. Each visualization session will start with a demonstration of one of the visualization tools (iVoLVER or Tableau). After this, we will ask you to use the tool to create visualizations of a small dataset. With your permission, we will record all your interactions with the visualization tools via screen captures – your face will not be visible in these. After the visualization session, we will ask you to fill out a questionnaire and we may ask you some questions about your experience with the tool. You will then do a second visualization session with the other tool, followed again, by filling out a brief questionnaire and answering some questions. At the end of the experiment, we will ask you some questions about your experience with both tools in comparison. With your permission, we will audio record this interview.

All data gathered during this experiment (i.e., data from questionnaires, audio recordings from interviews, and screen captures of your interactions with the visualization tools) may be used as part of research publications and presentations in **anonymized form**.

Will my participation be Anonymous and Confidential?
Only the researchers and the supervisor will have access to the raw data gathered during this experiment which will be kept strictly confidential. Data gathered during this study (statements from the interviews and questionnaires, and screen captures of your visualization process) will be fully anonymized prior to their use as part of publications and presentations. All data will generate anonymized versions of these materials (e.g. blurring your face and transcribing your spoken statements). We ask for explicit permission from you to share these anonymized versions of the video snippets or pictures in our publications. Please read carefully and answer the questions in the Consent form that accompanies this information sheet.

Storage and Destruction of Data Collected
The raw data collected during the study will be accessible by the researchers and the supervisor involved in this study only and stored for a maximum of 5 years. During its existence, raw data will be held securely in an office in the University of St Andrews' Department of Computer Science and in the securely encrypted data drives of the researchers.

What will happen to the results of the research study?

The results will be finalised by approximately 2018 and written up as part of my PhD Thesis and possibly as other research publications and presentations.

Are there any potential risks to taking part?

The risks of taking part should be no different from those you experience on a normal lecture or lab session.

Questions

You will have the opportunity to ask any questions in relation to this project before completing the Consent Form attached.

Consent and Approval

This research proposal has been scrutinised and been granted Ethical Approval through the University ethical approval process.

What should I do if I have concerns about this study?

A full outline of the procedures governed by the University Teaching and Research Ethical Committee is available at http://www.st-andrews.ac.uk/utrec/Guidelines/complaints
Contact details of the researchers are below.

Contact Details

Researcher:	Gonzalo Gabriel Méndez	
Contact Details:	ggm@st-andrews.ac.uk	
Researcher:	Uta Hinrichs	
Contact Details:	uh3@st-andrews.ac.uk	
Supervisor:	Dr. Miguel Nacenta	
Contact Details:	mans@st-andrews.ac.uk	Phone: +44 (0)1334 46 3265

D.2 Participant Consent Form – Coded Data

Participant Consent Form
Coded Data

Project Title
Studying the construction of visual encodings with iVoLVER and Tableau

Researcher(s) Name(s)

Gonzalo Gabriel Méndez
School of Computer Science
University of St Andrews
ggm@st-andrews.ac.uk

Dr. Uta Hinrichs
School of Computer Science
University of St Andrews
uh3@st-andrews.ac.uk

Supervisors Names

Dr. Miguel Nacenta
School of Computer Science
University of St Andrews
mans@st-andrews.ac.uk

The University of St Andrews attaches high priority to the ethical conduct of research. We therefore ask you to consider the following points before signing this form. Your signature confirms that you are happy to participate in the study.

What is Coded Data?
The term 'Coded Data' refers to when data collected by the researcher is identifiable as belonging to a particular participant but is kept with personal identifiers removed. The researchers retain a 'key' to the coded data which allows individual participants to be re-connected with their data at a later date if necessary. The un-coded data is kept confidential to the researchers and supervisors. If consent is given to archive data (see consent section of form) the participant may be contacted in the future by the original researchers or other researchers.

Consent
The purpose of this form is to ensure that you are willing to take part in this study and to let you understand what it entails. Signing this form does not commit you to anything you do not wish to do and you are free to withdraw at any stage and without providing a reason.
Material gathered during this research will be coded and kept confidentially, with only the researchers and supervisor listed above having access. All data will be securely stored in an office at the Computer Science Department of the University of St Andrews for a maximum of 5 years.
Please answer each statement concerning the collection and use of the research data.

I have read and understood the information sheet.	☐ Yes	☐ No
I have been given the opportunity to ask questions about the study.	☐ Yes	☐ No
I have had my questions answered satisfactorily.	☐ Yes	☐ No
I understand that I can withdraw from the study at any time without having to give an explanation.	☐ Yes	☐ No
I understand that my data will contain identifiable personal data, but that will be stored with personal identifiers removed by the researcher and that only the researchers and the supervisor listed above will be able to decode this information as and when necessary.	☐ Yes	☐ No
I understand that my data will be stored for a period of up to 5 years before being destroyed	☐ Yes	☐ No
I have been made fully aware of the potential risks associated with this research, and I am satisfied with the information provided.	☐ Yes	☐ No
I agree to take part in the study.	☐ Yes	☐ No

Part of our research involves taking video and audio recordings (of your screen and your interview, respectively). These recordings will be kept secure and stored with no identifying documents i.e. consent forms and questionnaires. With your consent, we would like to use this data in anonymised form as part of publications and presentations. We will transcribe all your verbal statements so that you cannot be identified based on your voice.

Audio and video recordings are highly valuable resources for future studies. We therefore ask for your additional consent to maintain this data for this purpose.

I agree to have my screen and interview video and audio recorded respectively.	☐ Yes	☐ No
I agree for my screen video and audio recorded material to be presented and published in anonymised form as part of this research.	☐ Yes	☐ No
I agree for all recorded data to be used in future studies.	☐ Yes	☐ No

Participation in this research is completely voluntary and your consent is required before you can participate in this research. If you decide at a later date that data should be destroyed we will honour your request in writing.

Name in Block Capitals ______________________________

Signature ______________________________

Date ______________________________

D.3 Demographics and Experience Questionnaire

«ID»

Study Constructing Visual Encodings

Demographic Information

1. What is your age? _________

2. What is your gender?

◯ Female

◯ Male

◯ Other

3. What is your current occupation?

◯ Undergraduate student

◯ Graduate student

◯ Professional

◯ Other

4. What is your area of study / professional expertise? ____

About your Visualization Experience

In the following questions, select the option that best represents your answer regarding your personal experience and background with visualizations. Please, select **just one option**.

5. How familiar are you with Information/Data Visualization?

◯ I do not know anything about information/data visualization at all.

◯ I started to explore visualization topics very recently. Before, I did not have any knowledge about it at all.

◯ I have come across information/data visualizations on the web and in magazines, but I have never considered the concepts and theory behind them.

◯ I have been actively following and reading about visualization related topics, but I do not have much experience with practically applying visualization concepts myself.

◯ I have been actively reading about the topic, and I have been creating visualizations myself for at least one years.

6. How familiar are you with reading and creating graphs and charts (e.g. bar charts, line graphs)?

◯ I do not really know what graphs and charts are.

◯ I have seen charts in newspapers and the like, but I have not paid much attention to them.

◯ I know graphs and charts from the newspapers and the web, and I know how to read them.

◯ I know how to read graphs and charts, and I have created some on my own a couple of times.

◯ I feel very comfortable at both reading and creating graphs and charts, and I do it often.

7. How often do you create information visualizations (e.g. bar charts, line graphs)?

- ◯ Never.
- ◯ I have sketched out concepts and/or data representations on paper before, but never created a visualization using computational tools.
- ◯ I have created some static visualizations (with Excel, for example).
- ◯ I regularly create basic charts and I have created at least one interactive visualization.
- ◯ I create interactive visualizations regularly as part of mystudies/research/job.

8. How would you rate your experience with the following visualization tools? Please, select just one option per tool.

	I do not know this tool.	I am aware of it, but I have never used it	I have used it once.	I am comfortable with it, but not proficient	I consider myself proficient.
Tableau Desktop	◯	◯	◯	◯	◯
iVoLVER	◯	◯	◯	◯	◯
Microsoft Excel	◯	◯	◯	◯	◯
D3.js	◯	◯	◯	◯	◯
Processing	◯	◯	◯	◯	◯
Google Fusion Tables	◯	◯	◯	◯	◯
Gephi	◯	◯	◯	◯	◯
Python / Bokeh / iPython Notebook	◯	◯	◯	◯	◯
Matlab	◯	◯	◯	◯	◯
R/Shiny	◯	◯	◯	◯	◯

D.4 Tool Interview Questions

«ID»

Study Constructing Visual Encodings

«Tool_1» Interview Questions

A. The Visualization & Satisfaction Level

1. Can you please explain a bit what your visualization(s) show and how they work?

2. So you answered you are generally pretty satisfied / unsatisfied with your visualizations.
   - **i.** <if satisfied> Can you explain what you think works particularly well?
   - **ii.** <if unsatisfied> Can you explain what you feel did not work so well in your visualizations?

B. Design Process

1. How would you summarize then the process of building visualizations with `<a_tool>`?
   - i. When you say `<something>`, you mean `<something_else>`?

2. How did you decide on the visual encoding of data in your visualizations?
   - i. I mean, what criteria did you use to map the data attributes to specific visual characteristics?
   - ii. Did you applied that strategy from the beginning or did you do something different and change it at some point?
   - iii. What did you do at first?
   - iv. Why did you change it?

3. Did you consider different designs before ending up with your final visualizations?
   - i. `<if_yes>`: Did you attempt to build any of them?
   - ii. Why did you discard them?
   - iii. `<if_no>`: So, you started with the goal of building a `<design_1>`, `< design_2>` and a `<design_3>`?
   - iv. How did you decide on particular types of visualizations?

4. Do you think `<a_tool>` allowed you to achieve all your design ideas?
   - i. So, you encounter limitations to express with the tool what you had in mind?
   - ii. Like what. Can you give an example?

5. Did you find that in `<a_tool>` you can always do what you wanted to?
6. Did the design process you have explained to me influence your level of satisfaction with your visualization?
   - i. If so, how does it influence your level of satisfaction?
   - ii. Why?
   - iii. Can you give examples?

C. Tool

1. Which features of this tool did you find **most useful**?
   - **i.** Do you mean the ….
   - **ii.** How come? / Why?
   - **iii.** Can you give examples?

2. Where there any features you encountered problems with?
   - **i.** Do you mean the ….
   - **ii.** How come? / Why?
   - **iii.** Can you give examples?

3. (**OPTIONAL for Tableau**) Were there any features that you felt were missing?

D.5 Tools Comparison Questions

Study Constructing Visual Encodings

Tools Comparison Questions

A. Differences

1. How did you experience the transition between Tableau and iVoLVER?

2. How do you think these two tools differ from each other?
    i. Can you give examples?
    ii. So, apart from <a_criteria>, do you think there is another difference
    iii. What about the type of visualizations you can build on each?
    iv. What about the process of creating visualizations?
3. If you compare the two tools, how would you summarize each of their benefits and limitations:

	BENEFITS	LIMITATIONS
IVOLVER		
TABLEAU		

B. Future Use

1. Now that you have used both tools, can you see yourself using them as part of your work/life activities?
    i. IVolver...
    ii. Tableau...
    iii. If "no" why not – do you use other tools already? Do you just not need to generate visualizations much?

2. If you think about the two tools, do you think they would be suitable for an educational scenario (InfoVis novices), I.e., for people just learning how to design and implement visualizations?
    i. iVolVER ...
    ii. Tableau ...
    iii. If you think they are suitable, can you explain what makes them suitable?
    iv. If you do not think they are suitable, can you explain why?

C. Preference

1. Overall, which tool would you prefer? Why?

Appendix E

Experimental Material Design Study

E.1 Participant Information Sheet

Participant Information Sheet

University of St Andrews

Project Title

Designing Language Elements and Interactions to Automate iVoLVER Visualizations

What is the study about?

We invite you to participate in a research project on how people design visuals and interactions to speed up the construction process of visualizations in iVoLVER, the **i**nteractive **V**isual **L**anguage for **V**isualization **E**xtraction and **R**econstruction.

This study is being conducted as part of my PhD Thesis in the School of Computer Science.

Do I have to take Part?

This information sheet has been written to help you decide if you would like to take part. It is up to you and you alone whether or not to take part. If you do decide to take part you will be free to withdraw at any time without providing a reason.

What would I be required to do?

You will be given a demonstration on how iVoLVER and Tableau visualizations are constructed. You will then use these two tools to create visual encodings of two small datasets. In this process, you will experience the different levels of automation of both tools. We will then ask you to design and sketch potential solutions to implement features in iVoLVER that allow users to speed up the process of building visualizations. With your permission, we will photograph your sketches as you work on them. In addition, at the end of the sketching exercise, we will record you while describing and discussing your designs in a group discussion where all the participants of the workshop will also participate. In this discussion, we will also ask your opinion on some of our own preliminary designs to implement automation in iVoLVER.

At the beginning of the experiment, we will ask you to provide some demographic details (e.g. age, gender) and to answer a set of multiple-choice questions (e.g. using numeric response scales) regarding your previous experience with visualization tools, sketches and design. We will use all the gathered data as evidence for discoveries in our research publications and presentations.

Will my participation be Anonymous and Confidential?

Only the researchers and the supervisor will have access to the raw data which will be kept strictly confidential. We will take notes, which will be anonymous. The results from the questionnaires will not be associated to your name in any publications; only the researchers will be able to connect your comments and answers to the questionnaires to your identity. If the videos and photos that we will take show your face, your voice, or identifiable parts of your body or your behaviour, we will generate anonymized versions of these materials (e.g. blurring your face and transcribing your spoken statements). We ask for explicit permission from you to share these anonymized versions of the video snippets or pictures in our publications. Please read carefully and answer the questions in the Consent form that accompanies this information sheet.

Storage and Destruction of Data Collected

The raw data collected during the study will be accessible by the researchers and the supervisor involved in this study only and stored for a maximum of 5 years. During its existence, raw data will be held securely in an office in the University of St Andrews' Department of Computer Science and in the securely encrypted data drives of the researchers.

What will happen to the results of the research study?

The results will be finalised by approximately 2018 and written up as part of my PhD Thesis and possibly as other research publications and presentations.

Are there any potential risks to taking part?

The risks of taking part should be no different from those you experience on a normal lecture or lab session.

Questions

You will have the opportunity to ask any questions in relation to this project before completing the Consent Form attached.

Consent and Approval

This research proposal has been scrutinised and been granted Ethical Approval through the University ethical approval process.

What should I do if I have concerns about this study?

A full outline of the procedures governed by the University Teaching and Research Ethical Committee is available at http://www.st-andrews.ac.uk/utrec/Guidelines/complaints
Contact details of the researchers are below.

Contact Details

Researcher:	Gonzalo Gabriel Méndez
Contact Details:	ggm@st-andrews.ac.uk
Researcher:	Uta Hinrichs
Contact Details:	uh3@st-andrews.ac.uk
Supervisor:	Dr. Miguel Nacenta
Contact Details:	mans@st-andrews.ac.uk Phone: +44 (0)1334 46 3265

E.2 Participant Consent Form – Coded Data

Participant Consent Form
Coded Data

University of St Andrews

Project Title

Designing Language Elements and Interactions to Automate iVoLVER Visualizations

Researcher(s) Name(s)

Gonzalo Gabriel Méndez
School of Computer Science
University of St Andrews
ggm@st-andrews.ac.uk

Dr. Uta Hinrichs
School of Computer Science
University of St Andrews
uh3@st-andrews.ac.uk

Supervisors Names

Dr. Miguel Nacenta
School of Computer Science
University of St Andrews
mans@st-andrews.ac.uk

The University of St Andrews attaches high priority to the ethical conduct of research. We therefore ask you to consider the following points before signing this form. Your signature confirms that you are happy to participate in the study.

What is Coded Data?

The term 'Coded Data' refers to when data collected by the researcher is identifiable as belonging to a particular participant but is kept with personal identifiers removed. The researcher(s) retain a 'key' to the coded data which allows individual participants to be re-connected with their data at a later date if necessary. The un-coded data is kept confidential to the researcher(s) (and Supervisors). If consent is given to archive data (see consent section of form) the participant may be contacted in the future by the original researcher(s) or other researcher(s).

Consent

The purpose of this form is to ensure that you are willing to take part in this study and to let you understand what it entails. Signing this form does not commit you to anything you do not wish to do and you are free to withdraw at any stage and without providing a reason.

Material gathered during this research will be coded and kept confidentially by the researcher, with only the researchers and supervisor listed above having access. All data will be securely stored in an office at the Computer Science Department of the University of St Andrews for a maximum of 5 years. Video snippets, photographs taken during the study sessions and data/statements that you provide via the questionnaires or during the group discussion may be used as part of research publications and presentations in anonymised form.

Please answer each statement concerning the collection and use of the research data.

I have read and understood the information sheet.	☐ Yes	☐ No
I have been given the opportunity to ask questions about the study.	☐ Yes	☐ No
I have had my questions answered satisfactorily.	☐ Yes	☐ No
I understand that I can withdraw from the study at any time without having to give an explanation.	☐ Yes	☐ No
I understand that my data will contain identifiable personal data, but that will be stored with personal identifiers removed by the researchers and that only the researchers and the supervisor listed above will be able to decode this information as and when necessary.	☐ Yes	☐ No
I understand that my data will be stored for a period of up to 5 years before being destroyed	☐ Yes	☐ No
I have been made fully aware of the potential risks associated with this research, and I am satisfied with the information provided.	☐ Yes	☐ No
I agree to take part in the study.	☐ Yes	☐ No

Part of our research involves taking video- and audio recordings as well as photographs. These recordings will be kept secure and stored with no identifying documents i.e. consent forms and questionnaires. With your consent, we would like to use this data in anonymised form as part of publications and presentations. We will blur any identifying parts in the video or photographs and fully transcribe verbal statements so that you cannot be identified based on your voice. However, you should know that there is still a chance that you are identified in these photographs and video recordings.

Photographs and video/audio are highly valuable resources for future studies. We therefore ask for your additional consent to maintain data and images for this purpose.

I agree to be video- and audio recorded and photographed during the study sessions.	☐ Yes	☐ No
I agree for my video, audio and photographic recorded material to be presented and published in anonymised form as part of this research. I understand that even in anonymised form video, audio and photos may still allow others to identify me.	☐ Yes	☐ No
I agree for all recorded data to be used in future studies.	☐ Yes	☐ No

Participation in this research is completely voluntary and your consent is required before you can participate in this research. If you decide at a later date that data should be destroyed we will honour your request in writing.

Name in Block Capitals ______________________________

Signature ______________________________

Date ______________________________

E.3 Demographics and Experience Questionnaire

Designing Language Elements and Interactions to Automate iVoLVER Visualizations

Demographic Information

1. What is your age? _________

2. What is your gender?

- ◯ Female
- ◯ Male
- ◯ Other

3. What is your current occupation?

- ◯ Undergraduate student
- ◯ Graduate student
- ◯ Professional
- ◯ Other

4. What is your area of study / professional expertise? ____

About your Visualization Experience

In the following questions, select the option that best represents your answer regarding your personal experience and background with visualizations. Please, select **just one option**.

5. How familiar are you with Information/Data Visualization?

- ◯ I do not know anything about information/data visualization at all.
- ◯ I started to explore visualization topics very recently. Before, I did not have any knowledge about it at all.
- ◯ I have come across information/data visualizations on the web and in magazines, but I have never considered the concepts and theory behind them.
- ◯ I have been actively following and reading about visualization related topics, but I do not have much experience with practically applying visualization concepts myself.
- ◯ I have been actively reading about the topic, and I have been creating visualizations myself for at least one years.

6. How familiar are you with reading and creating graphs and charts (e.g. bar charts, line graphs)?

- ◯ I do not really know what graphs and charts are.
- ◯ I have seen charts in newspapers and the like, but I have not paid much attention to them.
- ◯ I know graphs and charts from the newspapers and the web, and I know how to read them.
- ◯ I know how to read graphs and charts, and I have created some on my own a couple of times.
- ◯ I feel very comfortable at both reading and creating graphs and charts, and I do it often.

7. How often do you create information visualizations (e.g. bar charts, line graphs)?

- ◯ Never.
- ◯ I have sketched out concepts and/or data representations on paper before, but never created a visualization using computational tools.
- ◯ I have created some static visualizations (with Excel, for example).
- ◯ I regularly create basic charts and I have created at least one interactive visualization.
- ◯ I create interactive visualizations regularly as part of my studies/research/job.

8. How would you rate your experience with the following visualization tools? Please, select just one option per tool.

	I do not know this tool.	I am aware of it, but I have never used it	I have used it once.	I am comfortable with it, but not proficient	I consider myself proficient.
Tableau Desktop	◯	◯	◯	◯	◯
iVoLVER	◯	◯	◯	◯	◯
Microsoft Excel	◯	◯	◯	◯	◯
D3.js	◯	◯	◯	◯	◯
Processing	◯	◯	◯	◯	◯
Google Fusion Tables	◯	◯	◯	◯	◯
Gephi	◯	◯	◯	◯	◯
Python / Bokeh / iPython Notebook	◯	◯	◯	◯	◯
Matlab	◯	◯	◯	◯	◯
R/Shiny	◯	◯	◯	◯	◯

About your Interface Design Experience

In the following questions, select the option that best represents your answer regarding your personal experience and background with interface design and sketching. Please, select **just one option**.

9. How familiar are you with designing graphical user interfaces?

- ◯ I do not know anything about designing graphical user interfaces at all.
- ◯ I have come across graphical interfaces prototypes, but I have not worked on one myself.
- ◯ I started to design user interfaces very recently.
- ◯ I have been working on interface design of university projects for a while, but I do not have much real-world experience.
- ◯ I have been actively designing interfaces myself for at least one year.

10. Briefly describe the most relevant graphical user interface you have ever designed
(e.g., an academic project that involved a graphical/visual interface)

____

About your Sketching Habits

11. How familiar are you with sketching as part of your work, academic, or everyday life activities?

- ◯ I do not draw sketches as part of my work at all. Instead, I normally implement my ideas directly.
- ◯ I have sporadically interacted with someone else's sketches before trying to implement an idea.
- ◯ I have occasionally sketched ideas before implementing them, but it is not my usual approach.
- ◯ I sometimes use paper sketches when working. They seem to be useful, but not essential.
- ◯ I use both paper and digital sketches all the time to develop, evolve, and communicate my ideas before trying to implement them.

12. When you do sketch user interfaces, what kind of system/tool do you use?

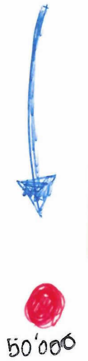
- ◯ Pen and paper
- ◯ Digital tools: ____
- ◯ Paper mockups
- ◯ Other: ____

E.4 Participants Sketches

P1

G1P1

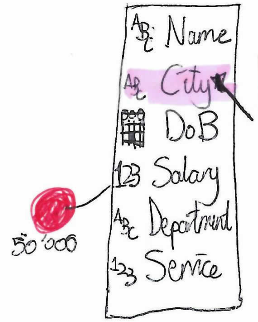
Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮



AUTOMATICALLY CREATE A MARK FOR EACH ROW

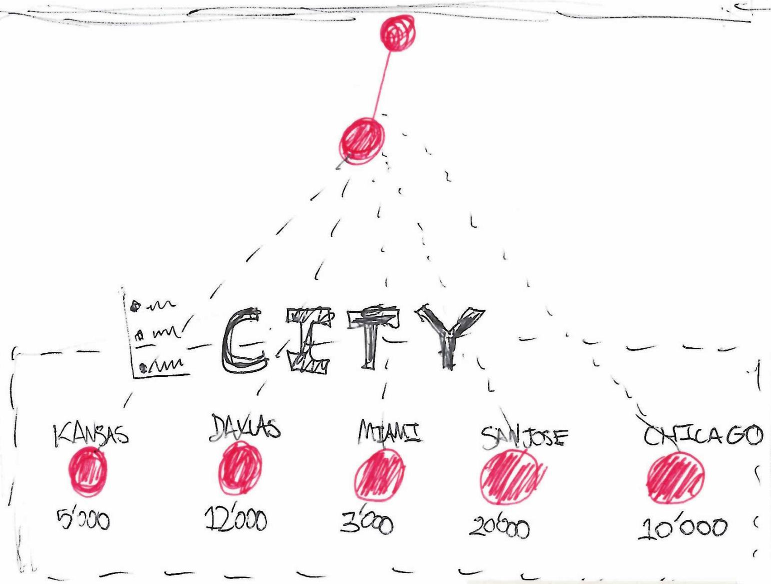
1

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮



Clicking the grouped mark shows all the possible attributes

2



Clicking on categorical fields creates markes grouped by category

3

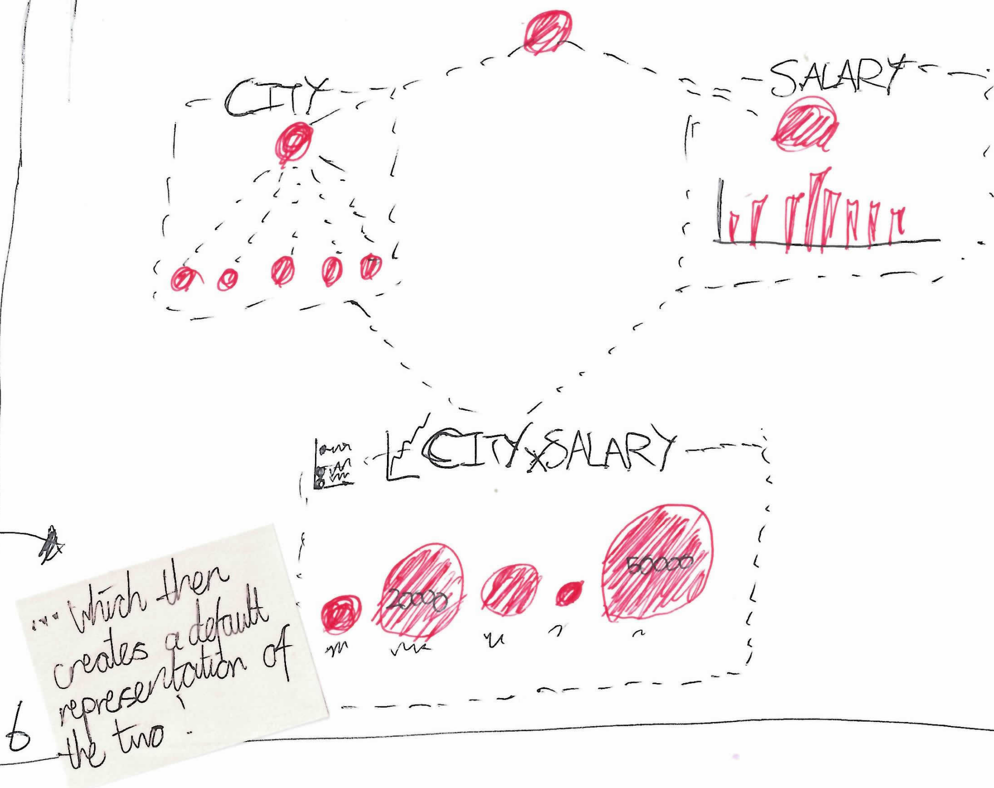
Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

Different groups of the same dataset can be created simultaneously

Note that categorical ( ) data is shown differently to numerical ( ) data

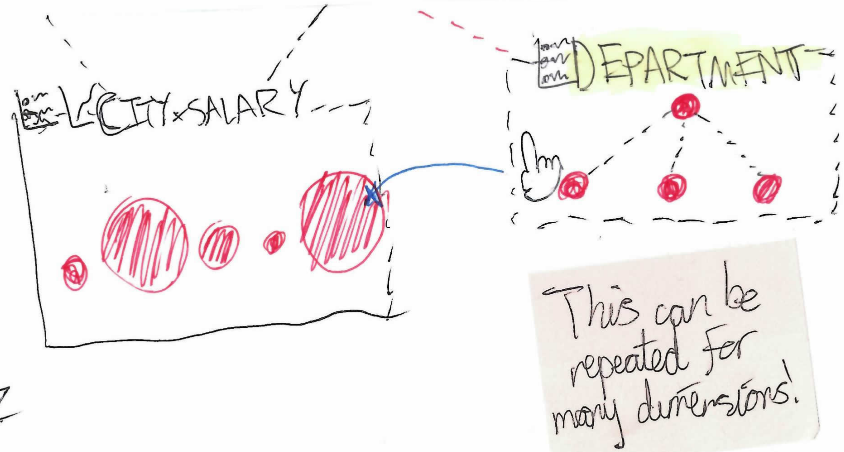
4

5



6

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

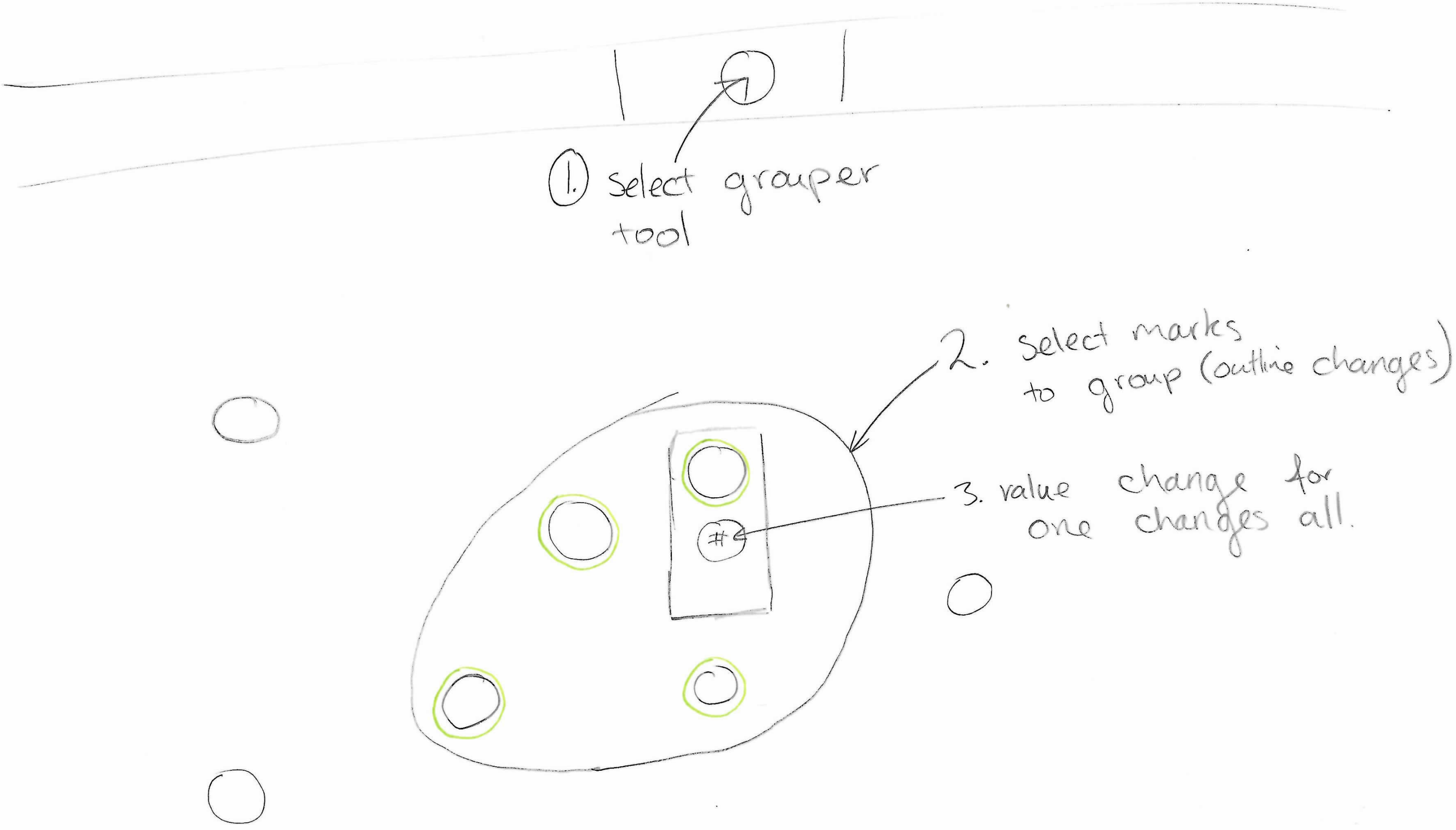


7

P2

G1 P2

"Grouper"



Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

"template tool"

data templater

mark templater

Step 1: create template table

3. assign data to template

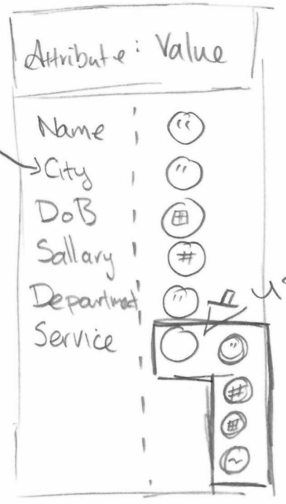
Steps
1. Create template of table
2. assign value types
3. assign values to mark template
~~4. Select dataset~~
5. generate

assumptions: an undo function would need to exist in case generation did not produce the correct result

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

Step 2: Assign value types

User lassos attributes and drags them to the templater



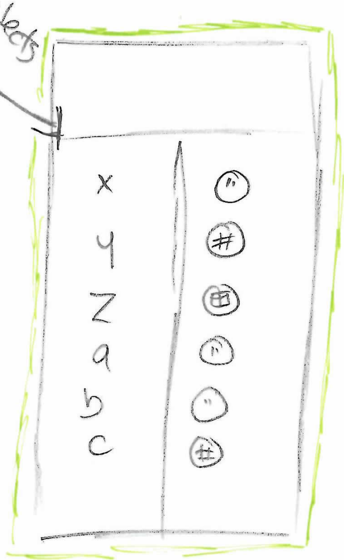
User clicks a value type + selects appropriate data type for each attribute

Step 3: Assign Data to Template

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

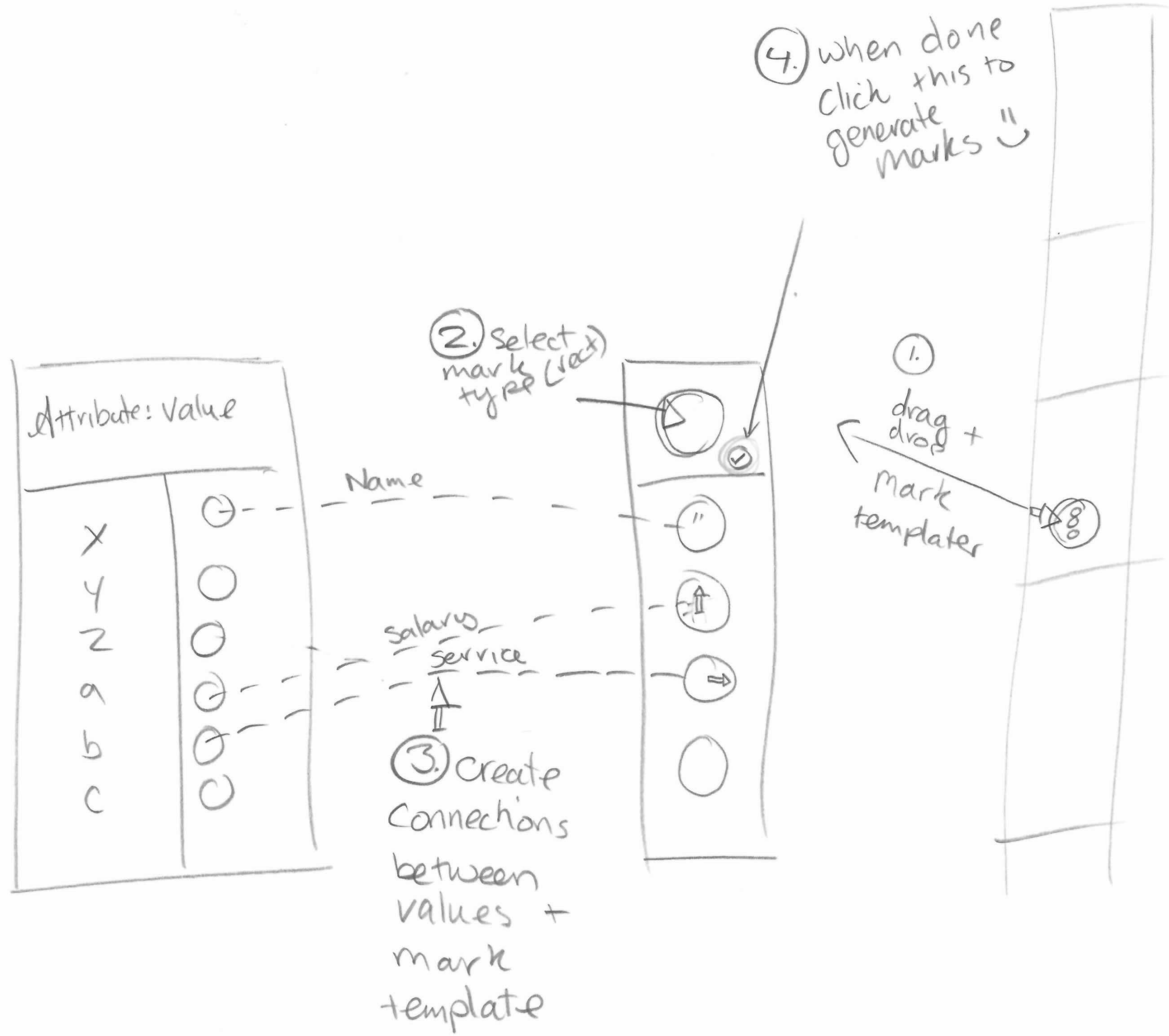
User selects data and drags it to template*

*in an ideal world done once, but could be done attribute by attribute



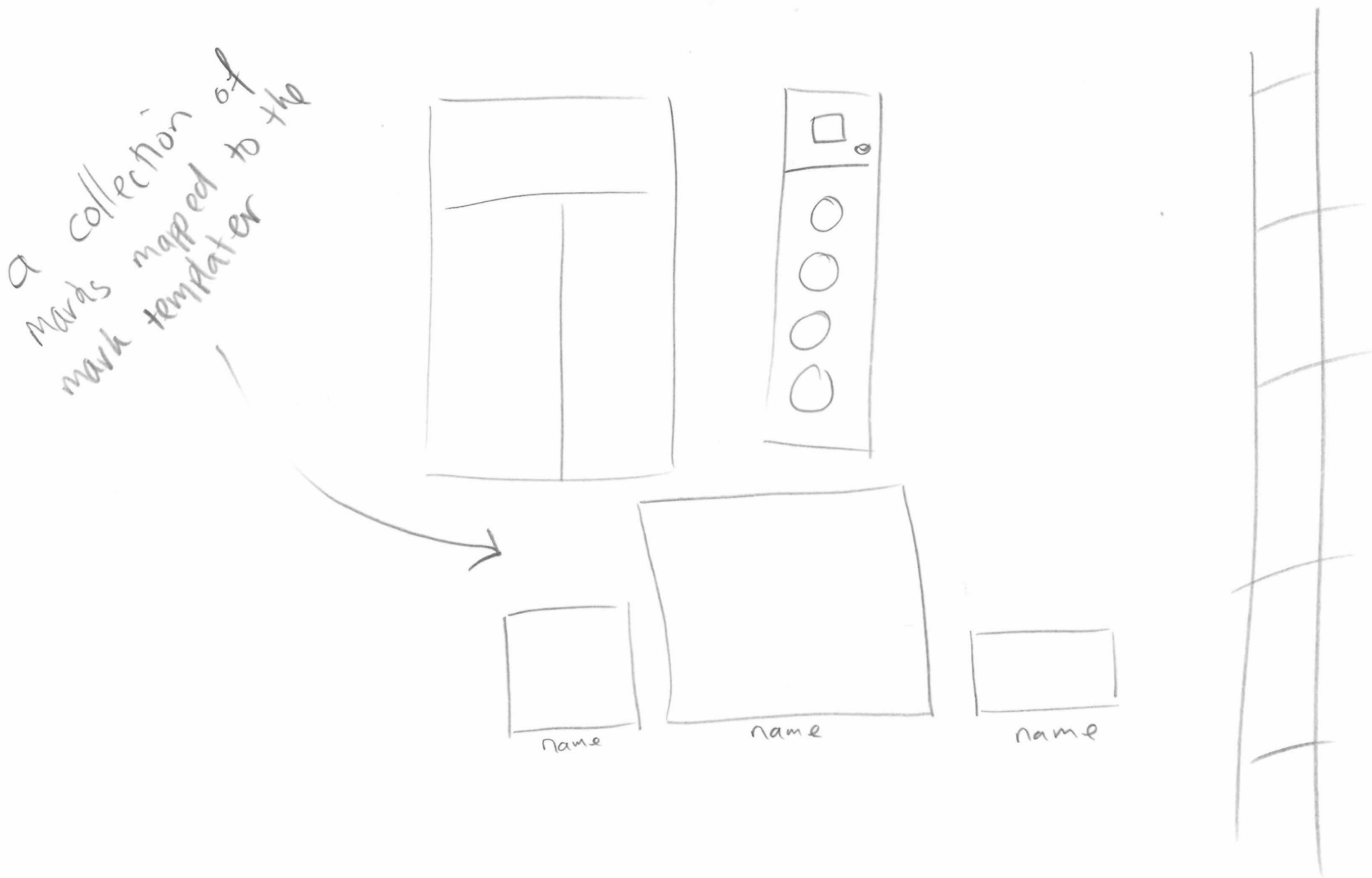
Outline changes to show data is held

Step 4: Assign Values to Mark Template
(+ step 5: Generate)



Presto chango! →

Step 6: Marvel in Your Genius (or hit undo)



P3

G1P3

Intelligent Extractor based on rows and columns

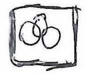
(1) The user selects the new widget on the menu

(2) The user uses ~~the~~ the tool to select the data he wants to work

(3)

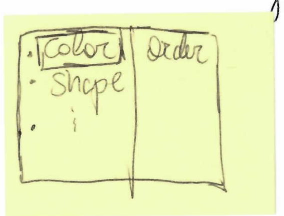
Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

"collection of cities"



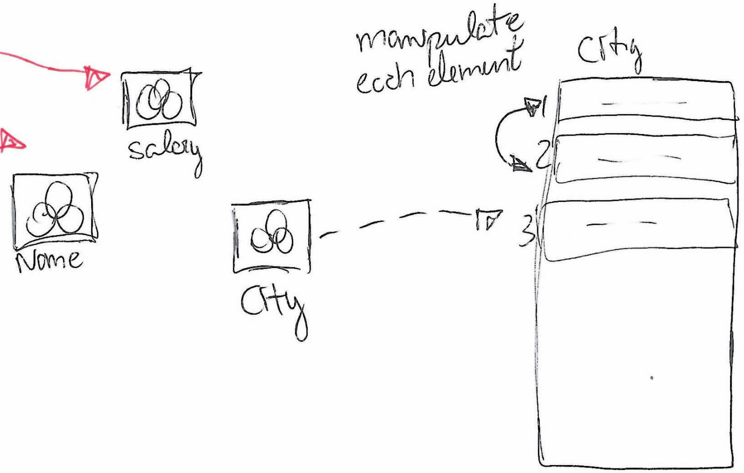
CITY

(4) Double click to change the properties of cities. This will ~~associate~~ link different colours to each element of this collection

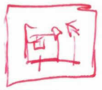


(5) Do the same with another row you want to work

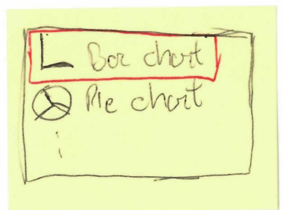
Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

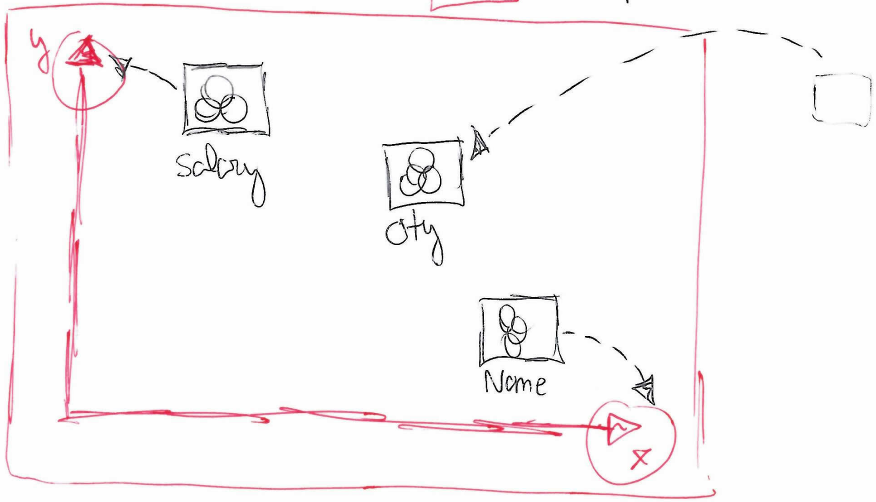


6 A new widget to automatically plot ~~the~~ relationship between ~~those~~ those collections



Double click to see all possible behaviours





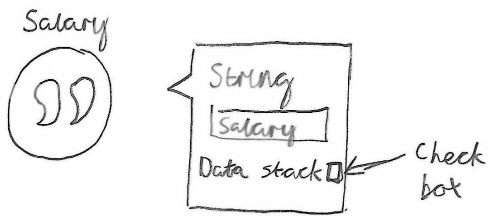
7

The user can easily drag 'n drop the collections to y if he wants to use it as y position

P4

G1P4

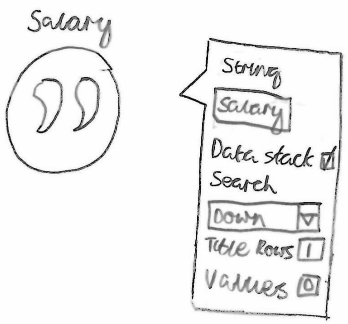
① After dragging string check the Data stack box



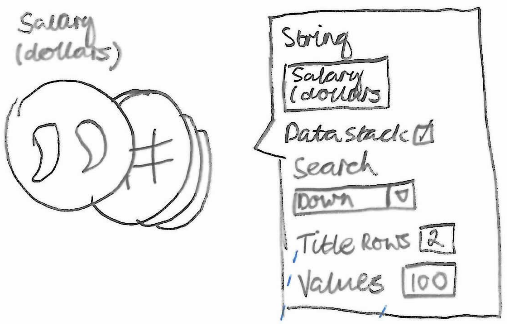
Drag

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

Next

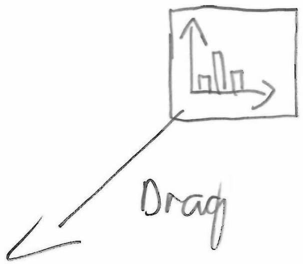


② Indicate direction of search, number of title rows and number of values. iVolve looks down and gathers the appropriate number of strings and values.

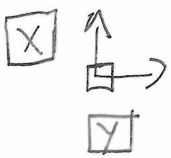


Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

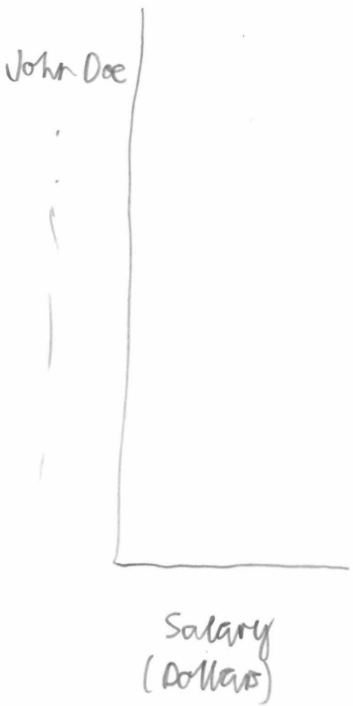
3 The string icon now has icons stacked behind it representing values



4 The user drags the graph icon to the canvas




Salary
(Dollars)
Name
Drag
Drag
Y
X


John Doe
Salary
(Dollars)

P5

G2P1

Data Set Widget

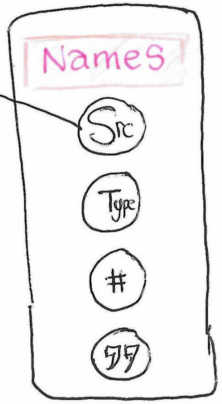
☆ Different Types of data Sets.

eg. Numeric/String/Date

Options

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	1
⋮	⋮	⋮	⋮	⋮	

Step 1: Creating the data Set Object



The value Holders:

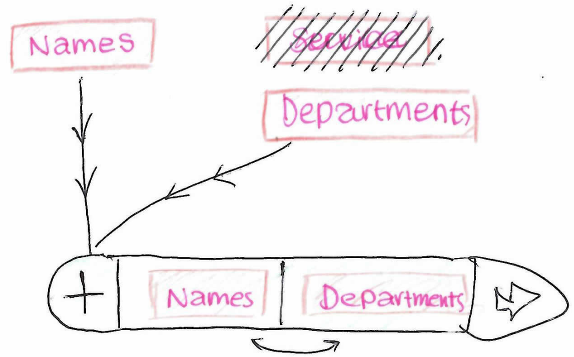
Src Provides the Source of the data (& whether Horizontal or Vertical)

Type Specify type of data if user wants to. Otherwise type doesn't have to be specified

\# Number of data to include. Default: max.

"" Name of data set

Step 2: Combining data Sets



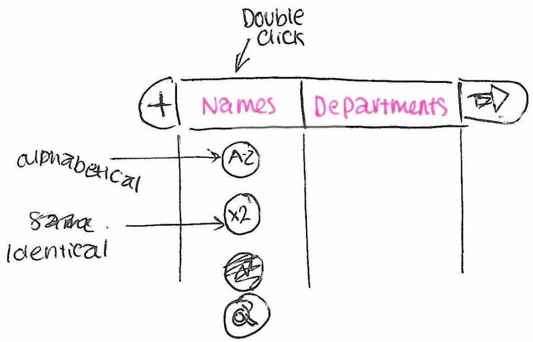
Can change order/remove etc.

maps names ↔ Department depending on Index

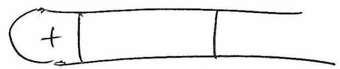
eg.

0:	Charles	Supplies
1:	Chris	Sales
2:	Rachel	marketing

Step 3: Drilling down the data:

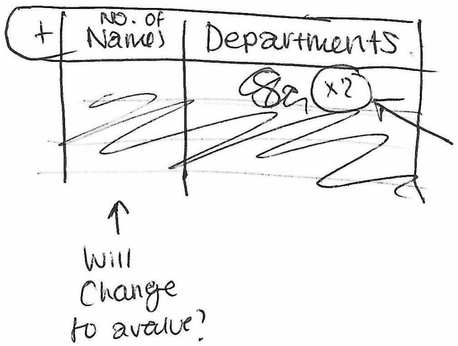


If Search is Selected:



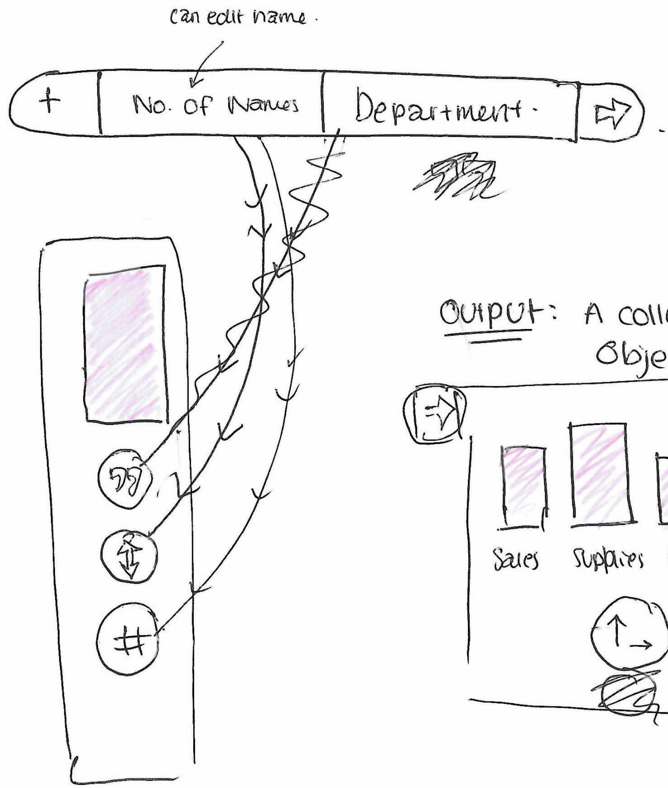
Will narrow down data.

If Identical is Selected:



Eg. Sales 20 ppl.

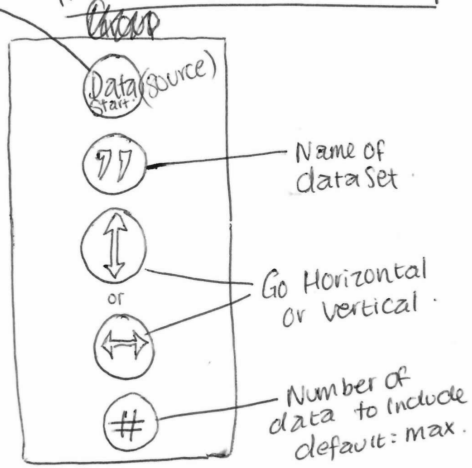
Step 4: Displaying the data

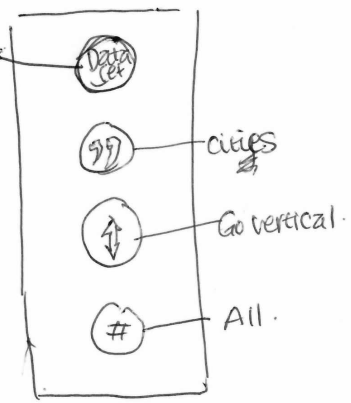


Output: A collection of Rectangle Objects.

Data Set Widget

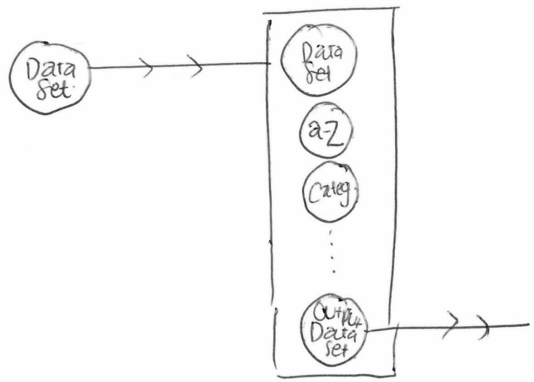
Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮





~~Eg construct.~~

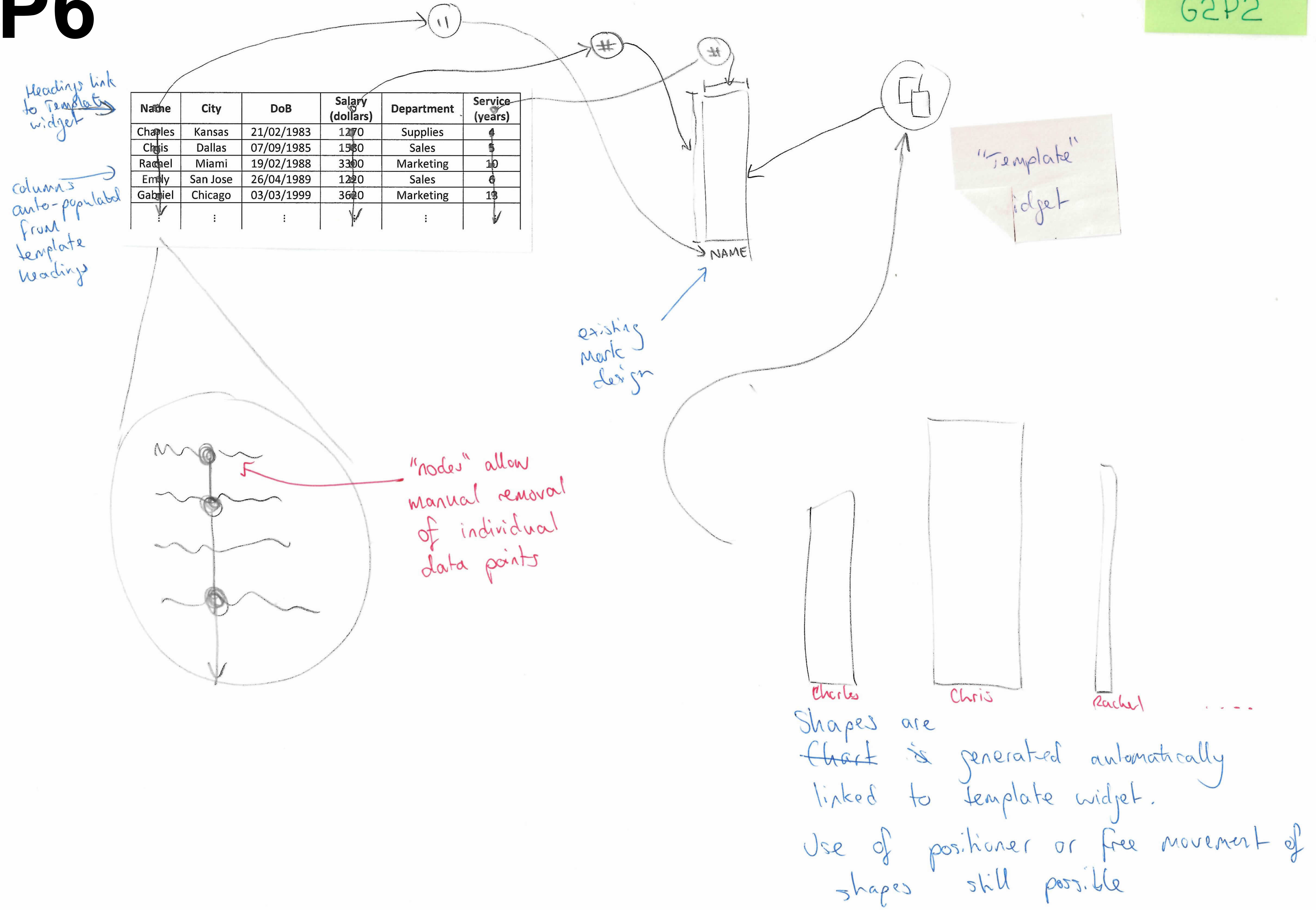
* Need a way to drill down data.



P6

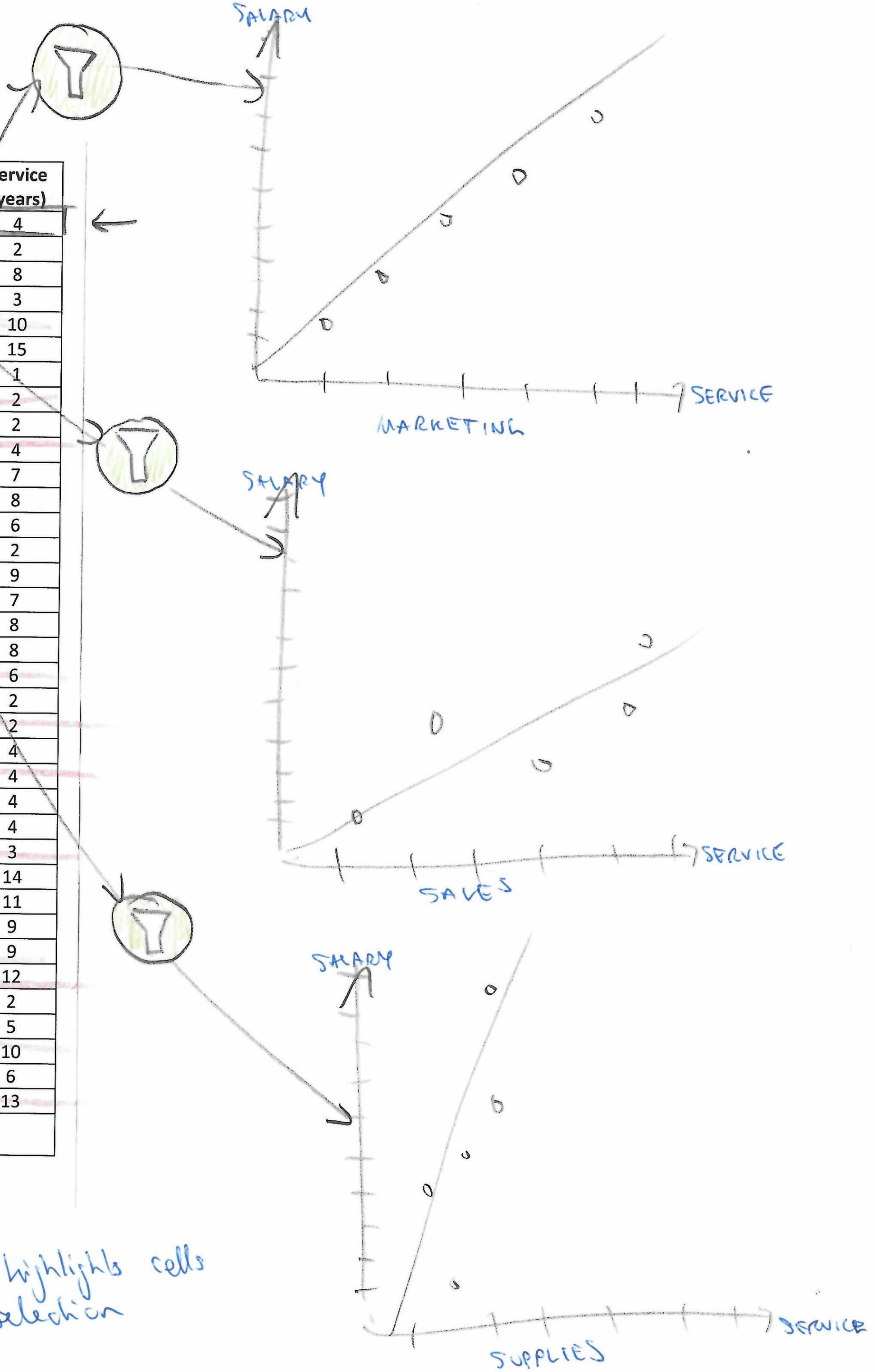
G2P2

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮



iVOLVER must be aware of cross-links in tabular data

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮



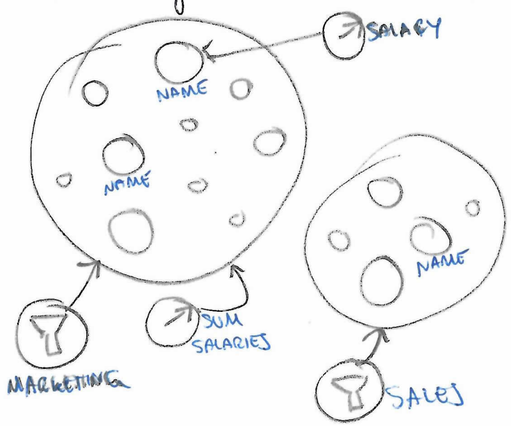
Related/linked data is lightly highlighted to illustrate connection

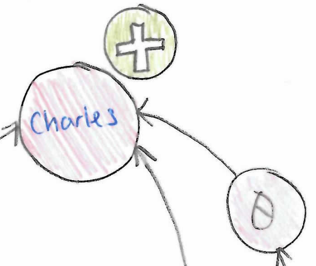
Filter highlights cells of selection

Data is linked across But filter widget allows auto-split or selection of subsets of data

Filter widget

Would work for so many kinds of visualisations!





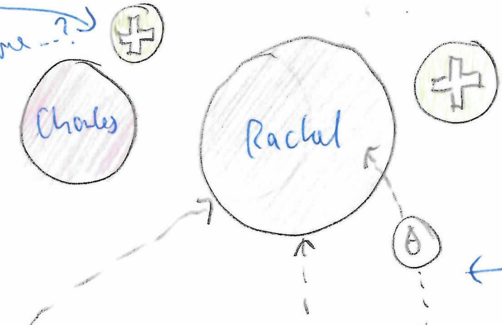
categorical info selects random colours. Continuous data provides gradient scale

Fields highlighted

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

⇒

May only appear on select... or all the time...?



links generated automatically

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

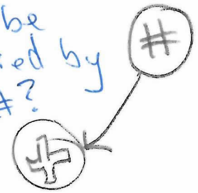
Clicking  allows duplication of a mark with new data following same structure.

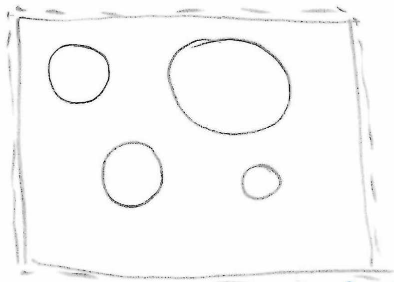
Next row may be auto selected but if the user chooses a different row it will be visualised instead

Plus popup

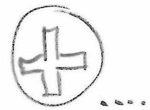
Options

Can be multiplied by a #?





Group selection offers further alignment options.
e.g. - randomly distributed
- close-packed
- overlapping



Extra dialogue on double-click for parameters (e.g. filter by data or select data items etc...)

P7

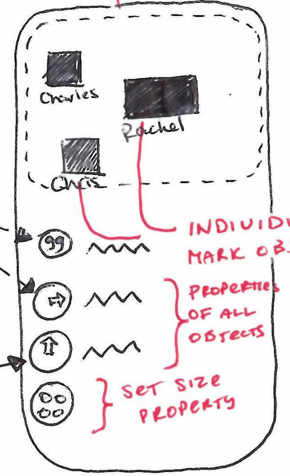
DESIGN #1

G2P3

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

1 CLICKING THE MARK SET BUTTON ALLOWS US TO PLACE A NEW MARK SET ENTITY

MARK SET OBJECT



INDIVIDUAL MARK OBJECTS

PROPERTIES OF ALL OBJECTS

SET SIZE PROPERTY

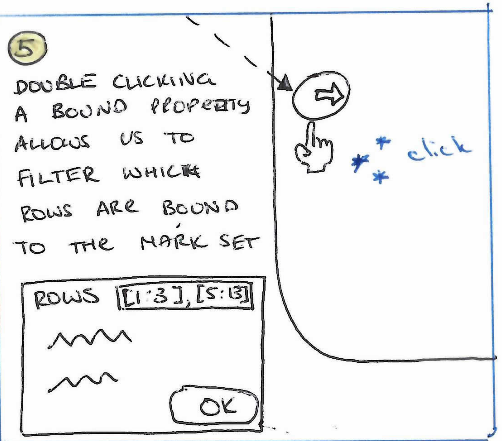
2 MARK SET OBJECTS ARE HOMOGENOUS. PROPERTIES OF ALL OBJECTS ARE ADJUSTED BY DOUBLE CLICKING THE SET.

3 WE CREATE BINDINGS BETWEEN COLUMN TITLES AND MARK SET PROPERTIES.

THIS IMPLICITLY ASSIGNS ONE ROW'S DATA TO EACH MARK, TAKING THE VALUE FROM THE COLUMN AND CREATING A BINDING TO THE ASSOCIATED MARK'S PROPERTIES

4 SIMILARLY, WE CAN BIND ~~COL~~ SINGULAR DATA VALUES TO MARK SET PROPERTIES. THE RESULT OF THIS IS THAT ALL MARKS ARE BOUND TO THE SAME VALUE.

5 DOUBLE CLICKING A BOUND PROPERTY ALLOWS US TO FILTER WHICH ROWS ARE BOUND TO THE MARK SET



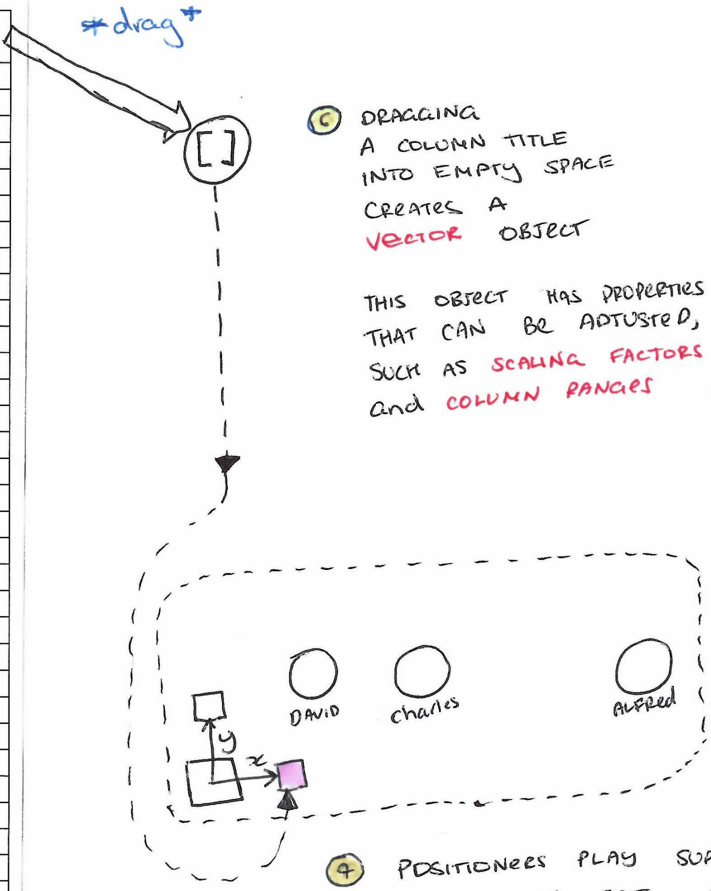
N-row data
→ need for a concept for a group of marks!

Issues:

→ Limitations on the types of visualisations that can be created
→ Need a new concept to allow for multiple marks to represent one data row
→ Mark Heterogeneity could be complicated to implement.

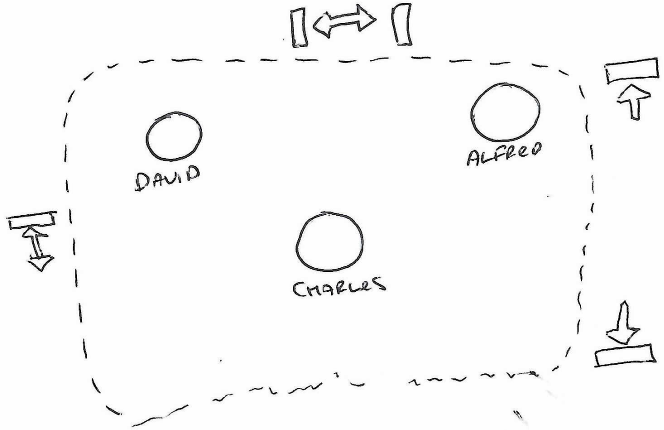
- THE GOAL IS TO ALLOW THE USER TO MANIPULATE N ROWS AT ONCE
- WE INTRODUCE THE CONCEPT OF A MARK SET, WHICH REPRESENTS A SET OF N MARKS
- WE CAN MANIPULATE PROPERTIES OF ALL MARKS IN THE SET AT ONCE BY DRAGGING COLUMN TITLES INTO SET OPTIONS

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮



(c) DRAGGING A COLUMN TITLE INTO EMPTY SPACE CREATES A VECTOR OBJECT

THIS OBJECT HAS PROPERTIES THAT CAN BE ADJUSTED, SUCH AS SCALING FACTORS and COLUMN RANGES



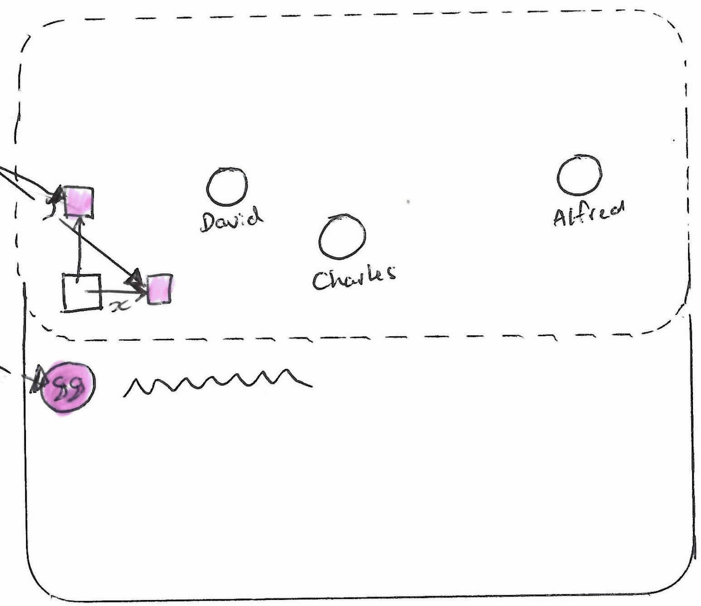
(8) WHEN NO POSITIONER IS PRESENT, MARK SET ELEMENTS CAN BE MANIPULATED ARBITRARILY BY DRAGGING THEM AROUND.

GROUP ALIGNMENT OPTIONS ARE ALSO AVAILABLE WHEN A MARK SET IS SELECTED, SO LONG AS A POSITIONER IS NOT PRESENT.

(9) POSITIONERS PLAY SUPER WELL WITH MARK SETS. IN THIS CASE, WE'VE DRAGGED A VECTOR OBJECT INTO THE x-AXIS OBJECT OF THE POSITIONER. EVERY ELEMENT IN THE SET NOW TAKES ON ONE VALUE FROM THE SERVICE COLUMN.

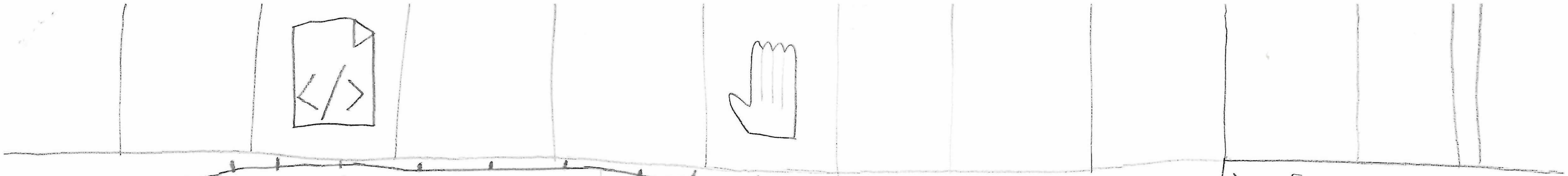
WE ADD POSITIONERS INTO A MARK SET BY DRAGGING. ONLY 1 POSITIONER IS ALLOWED PER MARK SET.

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮



P8

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮



Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

P9

G3P1

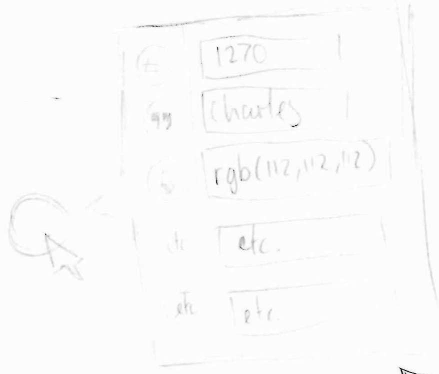
Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

Copy

① The user selects a retangle shape and places text "Charles"

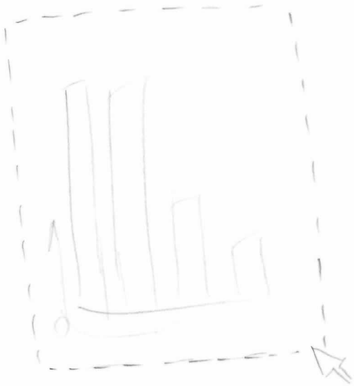
② Once copied, user is prompted to enter data, instead of re-dragging everything and double clicking

← New widget, copy

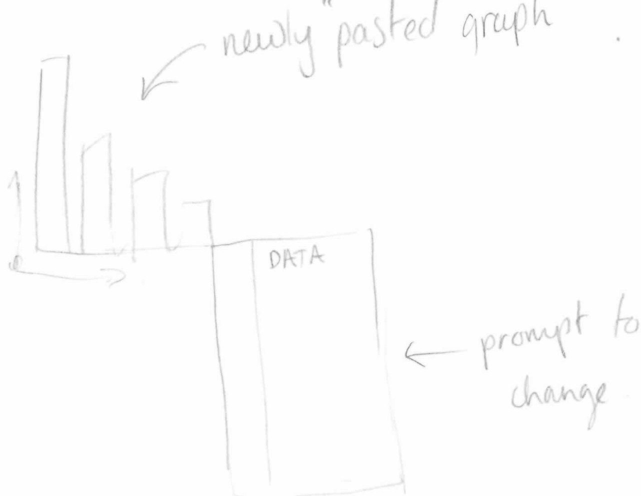


user has copied shape with existing data + is prompted to customise data to create similar marks.

③ Press & hold 'copy' option, allows users to "draw box"

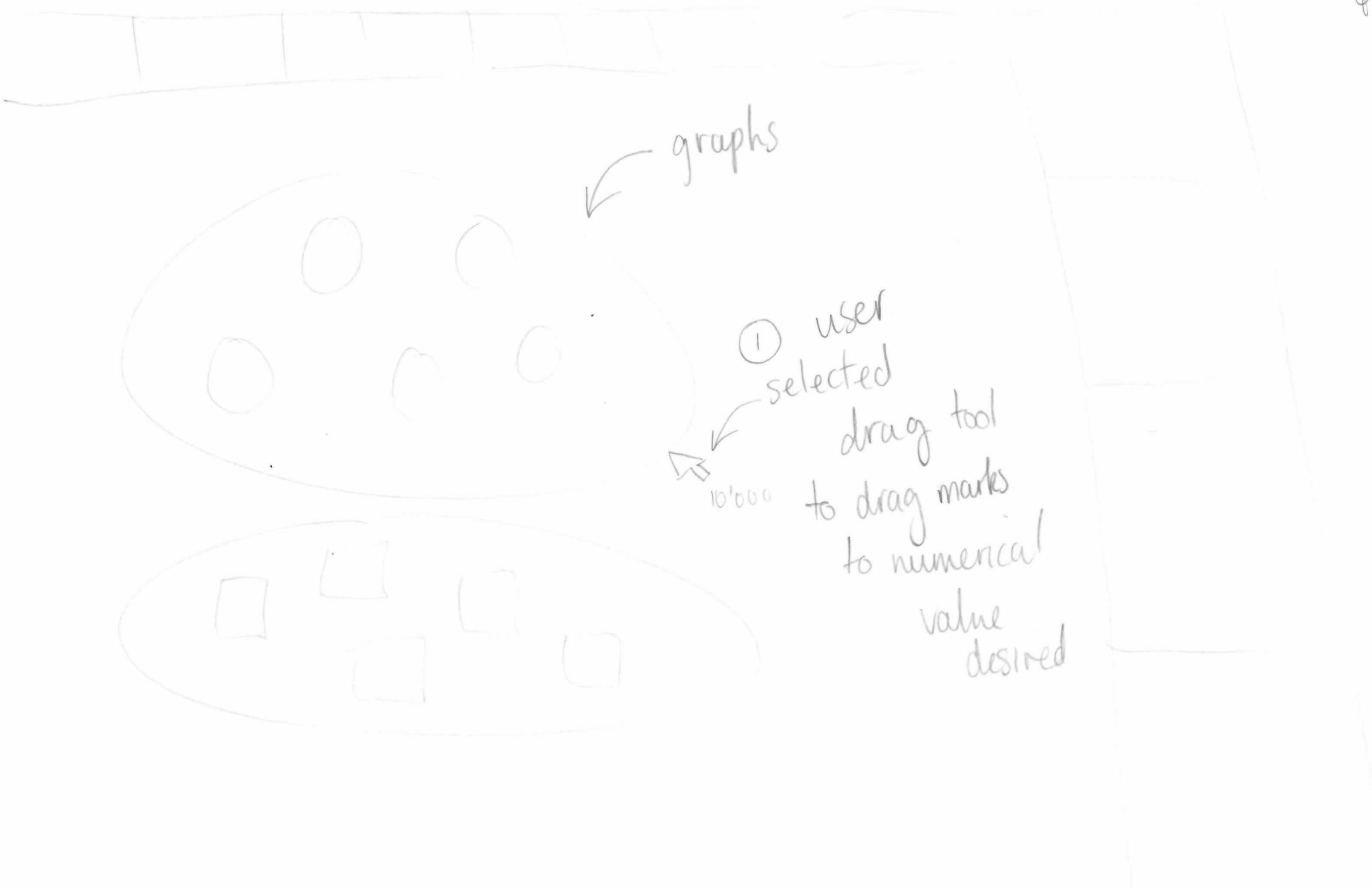


④ Once box is selected, the interior can be replicated but user will once again be prompted to correct data



Drag tool

(2) Large datasets are still entered but a drag tool makes sizing quick, & establishes numerical value automatically, and adjustable with the. drag tool

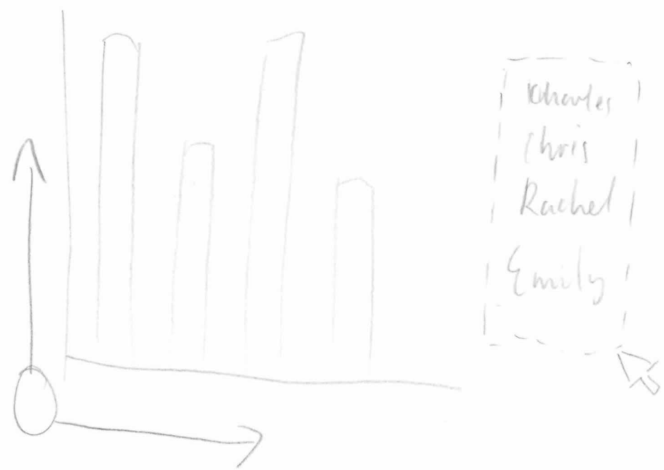


Select All

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

1 User selects all names or other category.

2 User can then drag the names onto a graph, together.



3 User can snap the data onto the graph, like the 99 option but on a bigger scale.

P10

G3P2

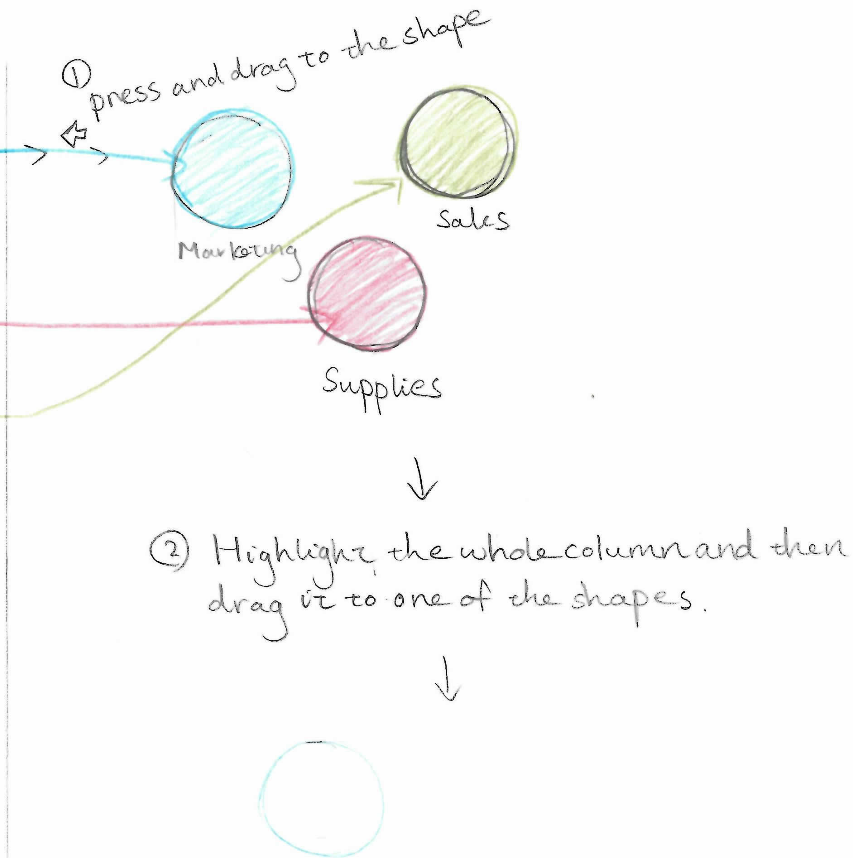
Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

① press and drag the whole column to highlight it. and release

② click on the highlighted column and drag it to the shape

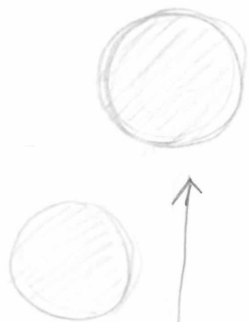
③ this creates a shape for each city.

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮



Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

① drag the whole column



P11

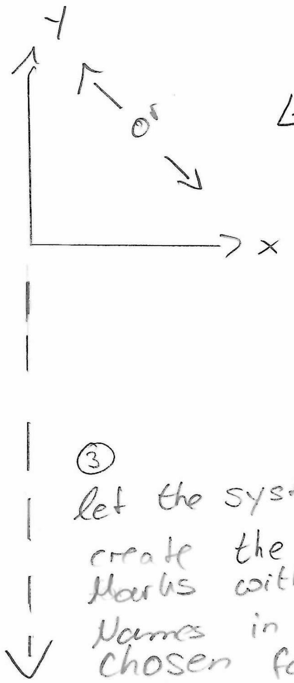
G3P3

> ~~

v Marks

① Drag a mark onto the category name

② Choose a direction/achse for the marks



Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

③ let the system create the Marks with Names in the chosen format just in one line

Charles David Alfred Alvin Andrew Alastair Fenton (...)

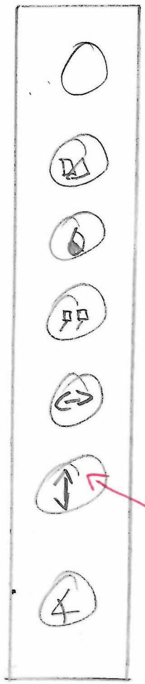
sheet ①

Tools →

(5) choose all marks

(4)

Charles | David | Alfred | Alvin | Andrew | (...)



(6) get a drop down menu for all chosen marks

(7) Choose another category name, drag and drop it to the field you want it

(part of step 7)

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

(8) a menu opens where you can choose the unit and factor (10^3 or 10^{10})

SALARY | dollars

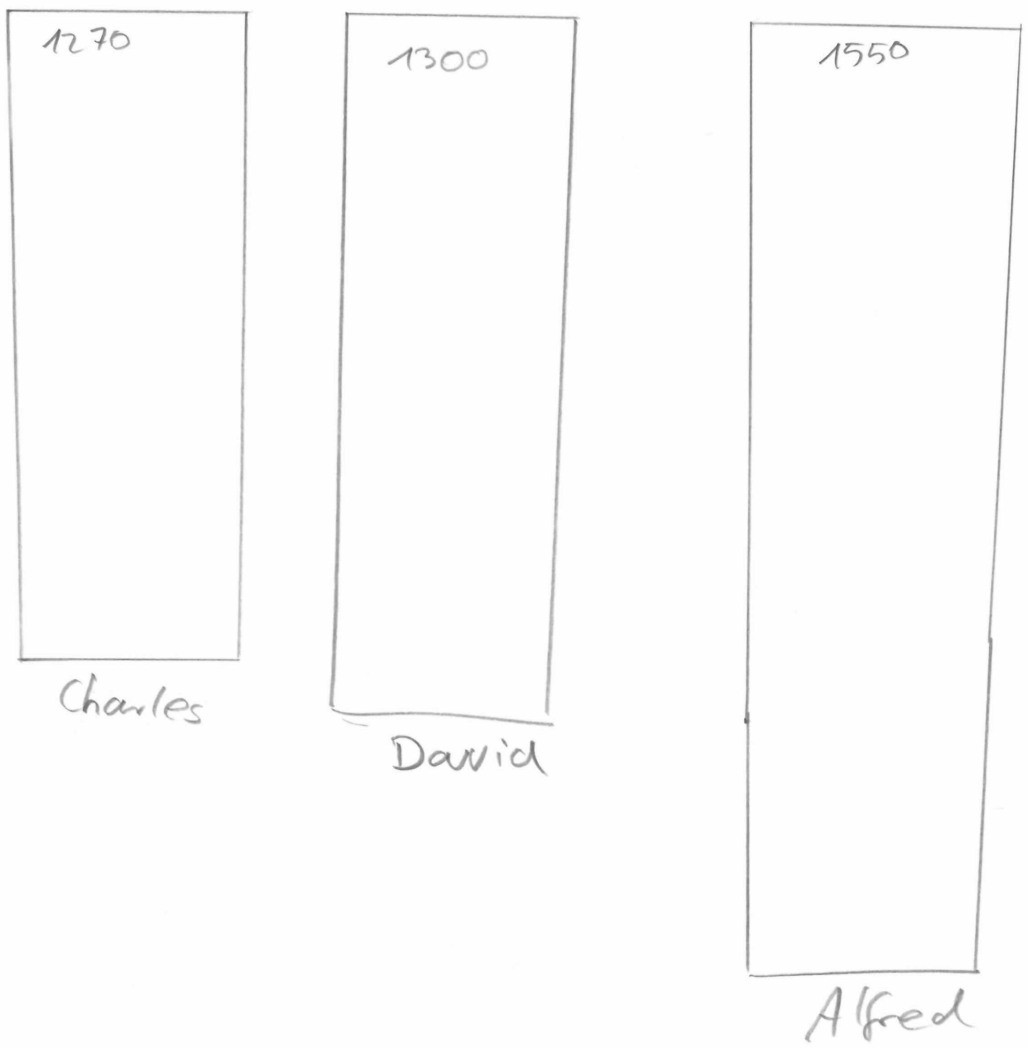
10^1

(9) The programm will combine all chosen marks with the given data in the same order they are provided and in the chosen format

sheet (2)

(10)

10

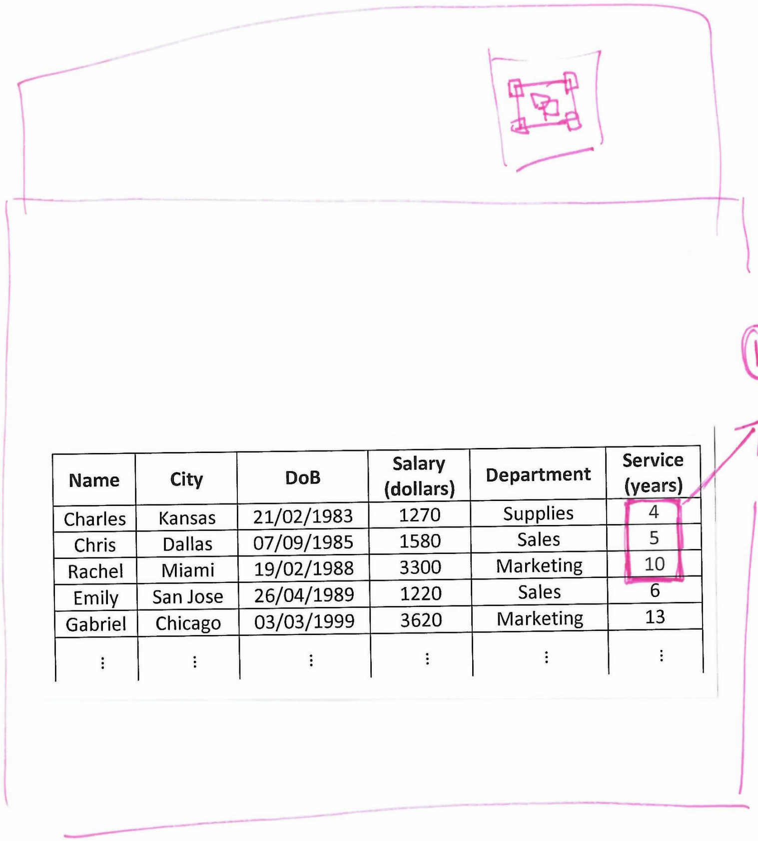


Sheet 3

P12

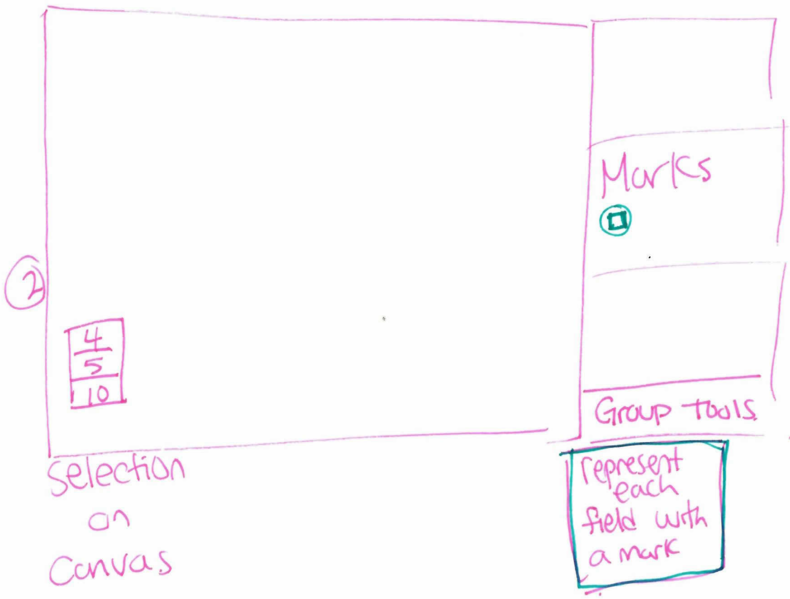
Selecting in Bulk.

G4P1



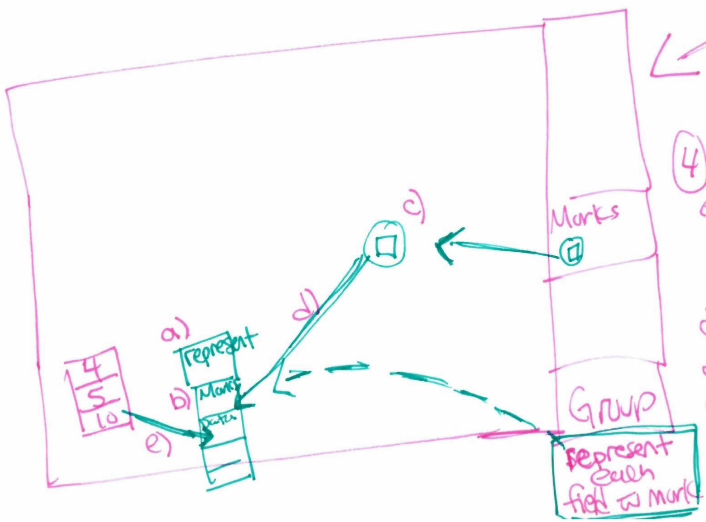
Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

1 select tool that enables you to select a section of data from a column.

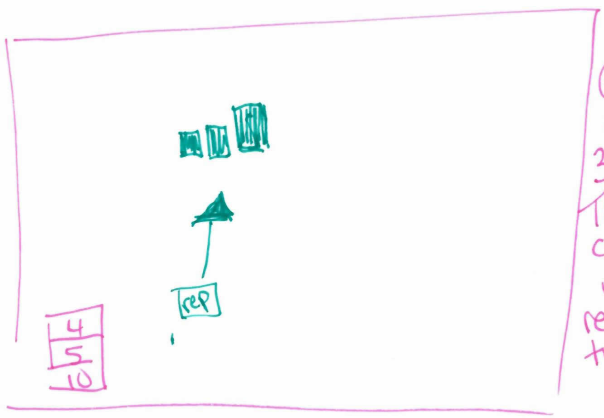


Selection on Canvas

3 ~~Hag~~ Having a menu similar to the Positioners Menu that would allow you to treat your selected data subset



4 a) Drag menu item on ~~canvas~~ canvas. ~~Screen~~ b) Double click for drop down.
c) Drag Mark on canvas
d) Affiliate Mark w represent
e) affiliate data extraction with Data field



5 Separated output of 3 ~~#~~ rows. These ~~po~~ marks can now be moved, but remain linked to the 'rep' relationship

Break Table Down.

Isolate Into separate entities

Name	City	DoB	Salary (dollars)	Department
Charles	Kansas	21/02/1983	1270	Supplies
Chris	Dallas	07/09/1985	1580	Sales
Rachel	Miami	19/02/1988	3300	Marketing
Emily	San Jose	26/04/1989	1220	Sales
Gabriel	Chicago	03/03/1999	3620	Marketing
⋮	⋮	⋮	⋮	⋮

b)

a)

Name

(Name column copied, but remains in original group as well)

Service

(Service now no longer in original group)

Copy

Separate

Highlight

Highlight

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

(1)

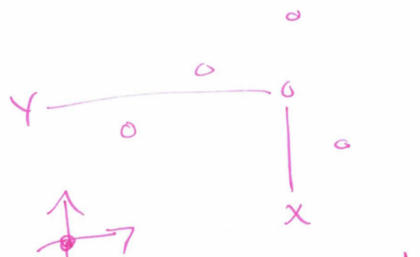
Highlight either selected ~~rows~~ rows or whole column..

Create Connections Between Table Rows/Columns

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮		⋮	⋮

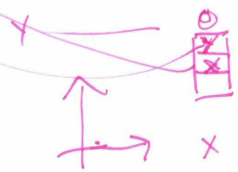
Double Click and Hold To Drag

(2)



Output of selected fields

(3)



This field remains editable

Treat whole Column ~~table~~

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

Select Column.

→ Create Number Range

→ Create Alphabetacal Range

→ Create Date Range

P13

- Titles, captions, labels (axes, values)
- Format painter
- Bulk select for mark creation / encoding. → Ctrl + click or selecting column header for whole column.
- ordinal colour scales
- create new columns in data set for calculated values.
- create colour from recently used.
- Sort data.
- Visual gridelines when dragging elements around canvas

MACRO

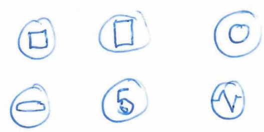
- Tripple click to select all / highlight all similar / 'like' items.
- record macros to automate actions. Make one visual / data encoding and then specify how many rows / which rows you want action repeated for.
- legends.
- start from a chart

G4P2

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11
Egerton	Poplar Hills	20/12/1965	3600	Sales	9
Erskine	Sweetwater	01/01/1990	1250	Marketing	9
Fletcher	Millbourne	01/10/1972	4500	Marketing	12
Forrest	Cambridge	03/01/1990	1250	Sales	2
Chris	Garfield	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

1. Use group select tool ( ) to select multiple parts of your data. Could be via drag + drop or Ctrl + click.
2. Once data selected, drag selection onto marks and release.
3. User then prompted with question about what to do. i.e. simply 1 mark per item, or 1 mark per item with visual encoding (size) drawn from another column of that item's corresponding row.

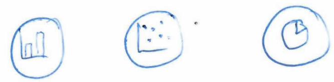
Marks



MACRO

1. Drag and drop a chart type onto the canvas.
2. User then prompted (prompts different for each chart type) to select axes labels, marks to plot (from data), columns to encode for size / height etc.

Charts



P14

(discrimination age) among the employees (salaries)

G4P3

1

Supplies

Sales

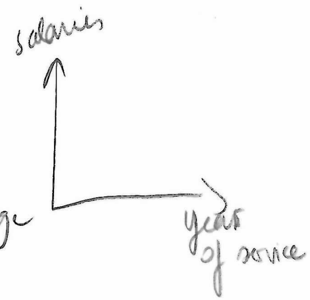
city

Average salary
Average Age
Average years of service

1

2

Supplies



Marketing

• By department

Average salary
Average Age
Average Years of service

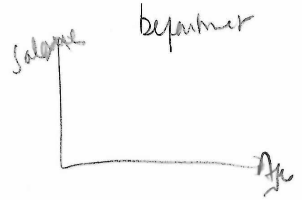
1) city

Average salary
Average Age
Average —

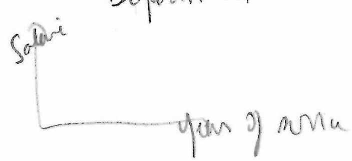
Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

2)

Department



Department



Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

select 1983 within DoB ?

transform 34 as a number

By department: average ?

By city: average ?

?

2

Age? "DoB" → Age transform a column ?

"recognize a number" 1983

2) Attempt Department

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
David	Guttenberg	20/12/1965	1300	Marketing	2
Alfred	Union City	01/01/1990	1550	Marketing	8
Alvin	Hoboken	01/10/1972	1245	Sales	3
Andrew	Kaser	07/09/1985	1220	Marketing	10
Alastair	Maywood	19/02/1988	1600	Sales	15
Fenton	Cudahy	26/04/1989	1980	Supplies	1
Benjamin	Somerville	03/03/1999	2500	Marketing	2
Broderick	Paterson	20/06/1985	1575	Supplies	2
Cary	Poplar Hills	07/06/1985	1580	Marketing	4
Chandler	Sweetwater	05/06/1983	3300	Sales	7
Conan	Millbourne	05/08/1982	1220	Sales	8
Crispin	Cambridge	21/02/1983	3620	Supplies	6
Culver	Garfield	20/12/1965	1245	Sales	2
Curtis	Jersey City	01/01/1990	1220	Sales	9
Craig	Chelsea	01/10/1972	1600	Sales	7
Dexter	Fairview	07/09/1985	1980	Supplies	8
Dirk	Long Beach	19/02/1988	2500	Sales	8
Donald	Lynwood	26/04/1989	1575	Marketing	6
Harrison	Millbourne	20/06/1985	1550	Supplies	2
Bruce	Cambridge	07/06/1985	1245	Marketing	2
Willian	Garfield	05/06/1983	1220	Supplies	4
Francis	Jersey City	05/08/1982	1600	Marketing	4
Drew	Kaser	03/03/1999	1980	Sales	4
Dudley	Maywood	20/06/1985	2500	Sales	4
Durwin	Cudahy	07/06/1985	1575	Marketing	3
Eaton	Somerville	05/06/1983	1580	Sales	14
Efrain	Paterson	05/08/1982	3300	Sales	11

1) highlight supplies

" supplies and yellow

[2] Repeat the tasks for marketing and sales]

2) choose services when yellow

" supplies and yellow and [,]

Name	City	DoB	Salary (dollars)	Department	Service (years)
Charles	Kansas	21/02/1983	1270	Supplies	4
Chris	Dallas	07/09/1985	1580	Sales	5
Rachel	Miami	19/02/1988	3300	Marketing	10
Emily	San Jose	26/04/1989	1220	Sales	6
Gabriel	Chicago	03/03/1999	3620	Marketing	13
⋮	⋮	⋮	⋮	⋮	⋮

interface